Protocoles de communications sécurisées par des séquences chaotiques. Applications aux standards de communications : IP via DVB-S, et l'UMTS

Daniel Caragata

UNIVERSITÉ DE NANTES
Ecole polytechnique de l'université de Nantes
France

UNIVERSITE DE PITESTI
Roumanie

———

ÉCOLE DOCTORALE

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET MATHEMATIQUES

Année 2011

PROTOCOLES DE COMMUNICATIONS SÉCURISÉES PAR DES SÉQUENCES CHAOTIQUES. APPLICATIONS AUX STANDARDS DE COMMUNICATIONS : IP via DVB-S, et l'UMTS

———

THÈSE DE DOCTORAT
Discipline-Spécialité : Electronique

Présentée
et soutenue publiquement par

Daniel CARAGATA

à l'Université de Pitesti, Roumanie

le 1er avril 2011, devant le jury ci-dessous

Président et Rapporteur	Alexandru SERBANESCU, Professeur, Académie Technique Militaire, Bucarest, Roumanie
Rapporteur	Jean-Pierre CANCES, Professeur, XLIM, Université de Limoges
	Bassem BAKHACHE, Maître de Conférences, Université Libanaise, Tripoli, Liban
Examinateurs	Safwan EL ASSAD, Maître de Conférences/HDR, IREENA, Polytech'Nantes
	Charles SHONIREGUN, Professeur, Infonomics Society, Basildon, Essex, Grande Bretagne
	Emil SOFRON, Professeur, Université de Pitesti, Roumanie
	Ina TARALOVA, Maître de Conférences, IRCCyN, Ecole Centrale de Nantes
	Ion TUTANESCU, Maître de Conférences, Université de Pitesti, Roumanie
Invités	Iulian RINCU, Maître de Conférences, Académie Technique Militaire, Bucarest, Roumanie
	Calin VLADEANU, Maître de Conférences, Université Polytechnique, Bucarest, Roumanie

Directeurs de thèse :
Safwan EL ASSAD, Laboratoire IREENA EA 1770
Emil SOFRON, Université de Pitesti, Roumanie

ED 503-121

A ma mère, qui me regarde de là haut, et à mon père.

Remerciements

Le travail présenté dans cette co-tutelle de thèse est le résultat d'une collaboration entre l'Ecole polytechnique de l'Université de Nantes et l'Université de Pitesti en Roumanie. Une thèse de doctorat est le fruit d'un travail collectif, pour cela je souhaite adresser mes remerciements aux personnes suivantes :

- Monsieur le Professeur S.TOUTAIN, Monsieur le Professeur J.SAILLARD, Monsieur le Professeur M. MACHMOUM, Directeurs successifs de l'IREENA pour m'avoir accueilli dans leur laboratoire.

- Monsieur le Professeur A. SERBANESCU de l'Académie Technique Militaire, Bucarest, Roumanie ainsi que Monsieur le Professeur J.P. CANCES de l'Institute de recherche XLIM, Limoges pour l'intérêt qu'ils ont porté à ces travaux, en acceptant de les rapporter.

- Monsieur S. EL ASSAD, Maître de Conférences HDR à l'Ecole polytechnique de l'université de Nantes et mon Directeur de thèse pour avoir cru en mon avenir en tant que chercheur, pour toute sa patience, sa pédagogie et toutes les choses que j'ai apprises avec lui. Son énergie, sa curiosité, son ouverture aux discutions et son dévouement pour la recherche scientifique m'ont beaucoup influencé.

- Monsieur le Professeur E. SOFRON, mon Co-Directeur de thèse pour avoir partagé avec moi sa grande sagesse.

- Monsieur le MCF I. TUTANESCU, mon encadrant pour toutes ses corrections, ses conseils et son soutien constant.

- Monsieur le Professeur A. SERBANESCU (encore une fois), et Monsieur le MCF I. RANCU pour m'avoir introduit au domaine de la sécurité de l'information basée chaos (quand j'étais encore étudiant), pour leur collaboration tout au long de ma thèse et pour leur soutien et participation à ma soutenance.

- Monsieur le Professeur C. SHONIREGUN de l'INFONOMICS SOCIETY, pour avoir participé à mon jury de thèse, et pour m'avoir donné l'opportunité de m'impliquer dans l'organisation des activités scientifiques. Aussi je remercie Madame G. AKMAYEVA de l'INFONOMICS SOCIETY pour son aide et sa collaboration.

- Monsieur le MCF B. BAKHACHE de l'Université Libanaise pour avoir travaillé à mes cotés, pour ses conseils, pour ses corrections minutieuses et pour avoir participé à mon jury de thèse.

- Madame le MCF I. TARALOVA de l'Ecole Centrale de Nantes, Monsieur le MCF C. VLADEANU de l'Université Polytechnique de Bucarest, Roumanie, pour avoir participé à mon jury de thèse et pour leur conseils et corrections.

- Madame S. CHARLIER, Monsieur M. BRUNET et toutes les personnes que j'ai côtoyées au sein du laboratoire pour leur soutien et aide tout au long de mon séjour au laboratoire.

- Anca Livia RADU, Cristian APOSTOL et Bogdan BOTEANU avec qui j'ai eu le plaisir de travailler pendant leurs stages de recherche Erasmus. Suivez vos rêves, vous pouvez faire tout ce que vous voulez.

- Mon collègue et ami Hassan à qui je lui souhaite sagesse et travail constant avec efficacité pour finir sa thèse.

- Mes amis qui m'ont aidé beaucoup quand j'ai eu besoin, avec qui j'ai partagé des moments excellents et qui ont fait que mon séjour à Nantes fait partie de mes expériences inoubliables. Je remercie aussi ma copine qui m'a supporté ces dernières années et qui m'a donné l'énergie de continuer dans les moments les plus difficiles.

Sommaire

Introduction générale

1. Contexte de l'étude et généralités

2. Communications IP par DVB-S sécurisées par des séquences chaotiques

Introduction générale

Introduction générale

L'importance et la pratique d'envoyer (porter) des messages secrets, ont été reconnues depuis l'antiquité. Cependant, durant des siècles, les méthodes utilisées étaient restées primitives, et leur mise en ouvre étaient limitées aux besoins de l'armée et de la diplomatie.

Aujourd'hui, l'émergence des nouvelles technologies de l'information et de la communication (communications sans fil, Internet, DVB-Satellite, messagerie électronique, vidéoconférences, commerce électronique, …), a favorisé l'explosion des échanges sur le plan mondial, et pose le problème, devenu public, de la sécurité de l'information échangée. Pour répondre à cette problématique, qui est évolutive, d'énormes travaux de recherche académique publique ont déjà été réalisés et d'autres sont en cours de réalisation.

Par ailleurs, depuis 1990, des travaux de recherche ont mis en évidence l'apport et l'intérêt d'utiliser des signaux chaotiques dans la sécurité des données échangées. En effet, des caractéristiques importantes des signaux chaotiques, telles que: un comportement apériodique à long terme, une large bande, une puissance constante, une efficacité aux trajets multiples, des bonnes propriétés cryptographiques, une reproductibilité à l'identique (déterministes), et une sensibilité à la clé secrète, incitent à leur utilisation dans tout système de communications sécurisées.

Le développement de nouveaux systèmes de sécurité de l'information basés sur les séquences chaotiques est un domaine de recherche relativement récent et en pleine expansion. Le nombre de publications à ce sujet est impressionnant et la recherche dans ce domaine est de plus en plus d'actualité, en jugeant par le nombre des conférences internationales dédiées à ce problème. L'essence des efforts théoriques et pratiques dans ce domaine, découlent du fait que les crypto-systèmes basés chaos sont plus rapides que les méthodes classiques, tout en assurant des performances de sécurité, au moins similaires. Notons au passage la nécessité de longue durée de validation de tels systèmes pour devenir des standards.

Les communications satellitaires ont été imaginés pour la première fois par Arthur C. Clarke dans l'article « Extra-Terrestrial Relays » publié en 1945 (Clarke, A.C., 1945). C'est à la fin des années 1950 que les premiers satellites de communications ont été envoyés en orbite. Aujourd'hui les satellites de communications sont utilisés principalement pour assurer des services de communications dans des endroits isolés (sommets de montagnes, plateformes pétrolières, déserts), dans des endroits frappés par une catastrophe (tremblement de terre, incendie, guerre), pour les véhicules à grande mobilité (trains de grande vitesse, bateaux, avions) et pour assurer une liaison entre deux points très éloignés (deux continents).

Les systèmes de communications DVB (Digital Video Broadcasting) offrent des services pour les réseaux de télévision câblée (DVB-C, DVB-C2), terrestre (DVB-T, DVB-T2), satellitaire (DVB-S, DVB-S2, DVB-RCS – DVB Return Channel via Satellite) et vers les appareils portatifs (DVB-H, DVB-SH – DVB Satellite to Handhelds) et sont très répandus en Europe, en Asie (sauf la Chine et le Japon), en Afrique et en Australie.

L'évolution des standards DVB, a permis leur utilisation dans les communications IP. Deux types d'encapsulation permettent la transmission des paquets IP par DVB satellitaire : l'MPE (Multi Protocol Encapsulation) et l'ULE (Unidirectional Lightweight Encapsulation). Cependant, l'ULE a des propriétés supérieures à l'MPE (G. Fairhurst 2003, Collini-Nocker B. 2004, Hong T.C. 2005). Pour cela, dans notre étude, nous ne traiterons par la suite que l'encapsulation ULE.

Un des problèmes qui reste ouvert pour l'encapsulation ULE, est la sécurité de l'information. L'encapsulation ULE, utilise un en-tête réduit au minimum pour ajouter un taux des données minimum et permet un mécanisme d'extension de l'en-tête pour envoyer des informations requises pour des services additionnels. Cependant, il n'y a pas une extension de l'en-tête, consacrée pour les services de sécurité. Pour pallier cet inconvénient, une proposition a été faite par (Cruickshank, H., 2008), mais la solution préconisée concerne seulement la structure de l'en-tête. Les aspects très importants tels la gestion des clés secrètes ou l'algorithme de chiffrement ont été laissés ouverts.

Pour les communications mobiles UMTS, lors de l'inscription auprès du réseau et à chaque changement de zone de localisation, des procédures de sécurisation de l'accès peuvent être exécutées pour assurer la sécurité des communications. La sécurité de l'UMTS présente des failles que nous mettons en évidence, puis nous proposons des solutions pour améliorer les services de sécurité.

Le développement des communications numériques a conduit à l'apparition d'un nouveau type d'information, l'information média, ayant des vulnérabilités particulières. Une de ces vulnérabilités est la possibilité de changer le contenu d'une image. Il y a un nombre important de logiciels qui permettent la modification du contenu des images: des personnes peuvent être effacées, rajoutées ou changées, etc. Pour cela, actuellement, la plupart des législations ne permet pas l'utilisation des images numériques comme preuve dans des procès. Pour résoudre ce problème, le tatouage des images numériques apporte une solution efficace.

Notre travail dans cette thèse tente de montrer comment les fonctions chaotiques permettent d'améliorer la sécurité des systèmes de communications IP par DVB satellitaire avec encapsulation ULE, et la téléphonie mobile de troisième génération, l'UMTS. Aussi, nous montrons l'efficacité des séquences chaotiques dans les algorithmes de tatouage fragile pour assurer l'intégrité des images JPEG.

Structure de la thèse

Dans le premier chapitre, nous introduisons les notions de base qui seront nécessaires à la compréhension de la suite de la thèse. Nous commençons par l'introduction des notions d'InfoSec et nous montrons que la protection de l'information est liée aux aspects théoriques, physiques cryptographiques et pratiques. Nous présentons les services de sécurité et les principaux types d'attaques auxquelles l'information peut être soumise. Ensuite, nous introduisons les notions de base utilisées dans le domaine de la sécurité de l'information et nous présentons brièvement les principaux types d'algorithmes cryptographiques à clé publique et à clé symétrique. Nous présentons aussi le lien entre le chaos et la cryptographie et nous discutons les problèmes liés à la représentation numérique du chaos et les remèdes proposés dans la littérature. La gestion des clés est un autre sujet important, traité dans l'introduction. Cependant, nous présentons seulement quelques notions de base, sachant que l'annexe A.3, traite ce sujet plus en détails. Enfin, nous présentons les divers types du tatouage numérique, leur structure et leur utilité.

Dans le deuxième chapitre, nous présentons une solution complète pour la sécurité des communications IP par DVB satellitaire avec encapsulation ULE, et ceci dans le cas des communications unicast et multicast. Notre solution contient une extension de l'en-tête ULE qui ajoute un nonce cryptographique, un protocole de gestion des clés multicouche, une fonction chaotique de génération des clés secrètes, un algorithme de chiffrement basé chaos et un PDU (Protocol Data Unit) utilisé pour le transport des données concernant les paramètres de sécurité. Nous analysons cette solution du point de vue du taux des données ajoutées et de la période d'utilisation de la clé master du système de gestion des clés multicouche. Aussi, nous présentons une analyse du générateur chaotique des clés secrètes et de l'algorithme de chiffrement proposé par l'équipe SCN du laboratoire IREENA travaillant sur la thématique sécurité des données.

Dans le troisième chapitre, nous traitons la sécurité des communications mobiles de troisième génération, l'UMTS (Universal Mobile Telecommunications System). Nous commençons par présenter la philosophie et l'architecture de la sécurité UMTS. La sécurité UMTS s'inspire de la sécurité GSM, elle utilise les mêmes grands principes (utilisation d'une carte à puce, d'une clé secrète K, d'un vecteur d'authentification) mais corrige les faiblesses de la sécurité GSM, liées aux algorithmes utilisés, et à l'authentification mutuelle entre l'utilisateur et le réseau. Par ailleurs, l'UMTS utilise une carte à puce plus évoluée que la carte à puce du GSM.

L'architecture de sécurité UMTS contient cinq domaines: niveau accès sécurisé au réseau, domaine réseau, domaine utilisateur, domaine application et visibilité et domaine configuration de la sécurité. Le niveau accès sécurisé au réseau est le plus important, car d'une part, il est spécifique dans le cas UMTS, et d'autre part, il est le maillon faible de tout réseau sans fil. A cet effet, nous avons concentré nos efforts sur cet aspect de la sécurité.

Dans un premier temps, nous présentons les trois composantes de l'accès sécurisé au réseau: l'identification des utilisateurs, la procédure d'authentification et d'établissement des clés (AKA – Authentication and Key Agreement) et l'établissement de la sécurité, ainsi que les fonctions de sécurité utilisées. Ensuite, dans une deuxième étape, nous proposons des améliorations de la sécurité UMTS.

L'identification avec l'identité permanente IMSI est le premier maillon faible de la sécurité UMTS que nous traitons. Nous présentons la solution proposée par Al-Saraireh (2006) et nous apportons trois améliorations à cette solution.

Le deuxième maillon faible traité, est la vulnérabilité de la clé K aux attaques cryptographiques sur la voie radio ou sur la carte à puce. Cette vulnérabilité à des conséquences néfastes sur la sécurité, car elle engendre la compromission totale des communications futures et passées. Pour cela nous proposons la protection des messages d'authentification et le renforcement de la clé K avec un mécanisme inspiré de la solution de protection de l'identité IMSI proposé par d'Al-Saraireh (2006).

Le troisième aspect que nous traitons, concerne la question de la confiance dans le réseau de service. Dans des cas spéciaux, tel que l'itinérance, le réseau de service appartient à un autre fournisseur que celui qui contrôle le réseau d'origine de l'utilisateur. Par sa structure, le réseau UMTS permet à un réseau de service mal intentionné ou n'utilisant pas d'algorithmes de chiffrement robustes de rendre les communications, d'un utilisateur donné, vulnérables. Nous proposons de petits changements dans les protocoles de choix des algorithmes et dans le TMSI, pour permettre au réseau d'origine et à l'utilisateur d'évaluer le comportement du réseau de service et ainsi savoir s'ils peuvent lui faire confiance.

Dans le quatrième chapitre, nous proposons un nouvel algorithme de tatouage numérique fragile, basé sur les séquences chaotiques, pour assurer le service d'intégrité des images JPEG.

Nous commençons par la description de l'algorithme de Wang qui nous a permis de développer une méthode de cryptanalyse de l'algorithme en question. L'algorithme de Wang utilise, pour générer l'information de tatouage ; une clé secrète, des séquences chaotiques dynamiques, et la valeur des coefficients JPEG quantifiés. Ensuite cette information est insérée dans les bits le moins significatifs des coefficients JPEG quantifiés. La cryptanalyse, que nous avons développée, utilise un nombre faible d'images tatouées pour trouver les paramètres nécessaires au processus de tatouage. Ces paramètres permettent à l'adversaire de tatouer n'importe quelle image, ou de modifier une image déjà tatouée.

Pour prouver l'efficacité de la méthode de cryptanalyse proposée, nous l'avons modélisée par les chaînes de Markov, puis simulé sous Matlab. Nous avons ainsi, calculé la probabilité de casser l'algorithme en fonction du nombre d'images tatouées utilisées. Nous avons montré qu'environ 20 images suffisent pour casser l'algorithme.

Ensuite, nous avons proposé un nouvel algorithme de tatouage fragile basé chaos des images, qui garde les avantages de l'algorithme de Wang, et assure la robustesse vis-à-vis de l'attaque développée.

1. Contexte de l'étude et généralités

1. Contexte de l'étude et généralités

La *cryptographie moderne* est la science des techniques mathématiques, permettant d'assurer des services de sécurité de l'information tels que : la confidentialité, l'intégrité, l'authenticité et la non-répudiation des données.

La cryptographie a une histoire très longue et fascinante, et sa première mise en œuvre heuristique par les égyptiens, remonte à approximativement 4.000 ans en arrière. Nous pouvons dire aussi que la cryptographie a changé le cours de l'histoire parce qu'elle a joué un rôle crucial dans beaucoup d'événements historiques, incluant les deux guerres mondiales. Les principaux praticiens de cet art étaient le personnel de l'armée, des services diplomatiques et gouvernementaux. La cryptographie était utilisée comme moyen de protection des secrets nationaux et stratégiques.

De nos jours, les techniques et moyens utilisés pour assurer la sécurité de l'information ont énormément évolués. La prolifération des moyens de communications de tout types (ordinateurs, réseaux d'ordinateurs, téléphonies mobiles, etc.) a favorisé l'explosion des communications sur le plan mondial et a introduit aussi, la problématique de la sécurité des flux des données échangés.

1.1. Sécurité de l'information: InfoSec (Information Security)

1.1.1. Information

Les sciences utilisent trois concepts de base: l'énergie, la matière et l'information. L'énergie est définie comme la capacité d'un système technique, biologique, etc. d'effectuer un travail. Elle existe sous différentes formes : énergie mécanique, énergie thermique, énergie chimique, et énergie de champs : gravitationnel, électrique, magnétique, etc.

La matière est la qualification de toutes les substances ayant une réalité tangible, opposée à l'énergie et le vide. La matière occupe de l'espace, et sa masse, est un élément intrinsèque.

L'information est soit stockée sur un support matériel ou portée par un signal physique. La notion de signal est très complexe. Il existe dans la nature, un nombre très varié de sources d'informations tout genre. Un signal peut être défini comme le moyen physique pour porter de l'information sur la présence ou l'évolution d'un système physique. Le *modèle mathématique* du

signal, représente une relation fonctionnelle dont l'argument est le temps *s(t)* ou le temps et l'espace *s(t, z)*.

La notion de *données* consiste dans la valeur mathématique ou logique obtenue à partir d'une mesure du signal par un dispositif tels que: oscilloscope, analyseur de spectre, voltmètres, ampèremètre, etc.

Une grande partie des systèmes de communications utilisent les signaux radio pour porter l'information. Dans ce contexte, ce n'est pas seulement les utilisateurs autorisés qui peuvent accéder à l'information mais des intrus aussi. Pour empêcher l'accès des intrus, la protection de l'information est nécessaire. Depuis toujours, une compétition se déroule entre celles et ceux qui veulent protéger le secret de leur communication et celles et ceux qui veulent accéder d'une manière non légal au secret en question.

1.1.2. Interconnexion des réseaux

Le besoin en communications des personnels au sein d'une même entreprise, institution, organisation a conduit à l'apparition des réseaux locaux de communication. Aussi, le besoin de communiquer entre entreprises, organisations, et instituions, a amené ces réseaux à être interconnectés, et ont donné naissance à des réseaux étendus nationaux et même globaux, tels que le réseau de téléphonie mobile et l'Internet. De nos jours, de secteurs stratégiques tels que les secteurs: énergétique, de distribution de gaz, de transport, financier, militaire, etc, dépendent des réseaux d'ordinateurs et de communications.

Lorsqu'un réseau est connecté à un autre, le risque d'attaques passives ou actives sur le flux de données qui transitent devient très important. Le réseau le plus connu et le plus vulnérable est l'Internet. Chaque réseau a ses propres risques, mais les réseaux connectés à l'Internet sont beaucoup plus exposés que les réseaux isolés. Pour cela, des nouvelles solutions de protection sont toujours recherchés.

1.1.3. InfoSec, ComSec (Communications Security) et ses composantes

InfoSec, comprend toutes les mesures, procédures ou contrôles qui offrent un niveau de protection acceptable contre la compromission, la modification ou la destruction intentionnelle ou involontaire des données stockées ou transportées par un système de communication.

La sécurité de l'information est une préoccupation de recherches scientifiques avec applications en communications et en systèmes de traitement des données. L'assurance sur la sécurité de l'information est un aspect important pour toutes les parties qui envoient des données dans un système de communication.

InfoSec, a trois composantes: la sécurité des communications (ComSec – Communications Security), la sécurité des ordinateurs (CompuSec – Computer Security) et la sécurité des réseaux (NetSec – Network Security). Donc, ce n'est pas seulement la sécurité des données qui fait partie d'InfoSec, mais aussi la sécurité des liaisons entre différents éléments du réseau et l'assurance du service.

InfoSec, vise à protéger les données contre la:

- Détérioration de l'intégrité physique et logique des liaisons;
- Modification des données;
- Réception non-autorisé.

InfoSec, doit tenir compte de tous les facteurs qui peuvent influencer de manière négative la transmission ou le traitement de l'information. Ceci permet de choisir la stratégie correcte qui offre un niveau élevé de protection de l'information avec une efficacité accrue.

ComSec, contient l'ensemble des mesures qui empêchent les personnes qui ne sont pas autorisées, à accéder au contenu des données communiquées ou à écouter les communications transmises. Les éléments les plus importants du ComSec sont :

- EmSec (Emission Security), sécurité de l'émission;
- TranSec (Transmission Security), sécurité de la transmission
- CryptoSec (Cryptographic Security), sécurité cryptographique.

Nous représentons les éléments du ComSec, dans la figure 1.1. Nous remarquons que l'EmSec, le TranSec et le CryptoSec sont des couches consécutives qui entourent l'information transmise.

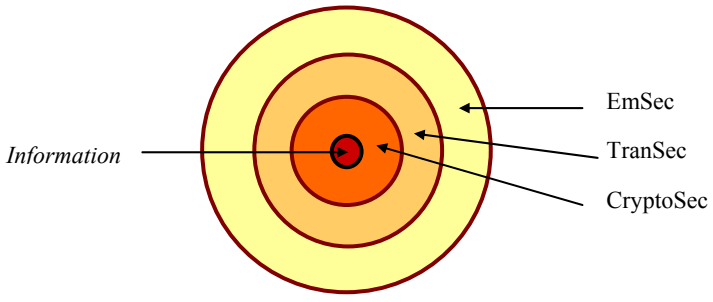


Fig. 1.1: Représentation graphique des éléments du ComSec.

EmSec, contient toutes les mesures qui empêchent les personnes non autorisées d'obtenir de l'information utile à partir de l'interception et l'étude des émanations compromettantes. Les émanations compromettantes sont des signaux involontaires qui, si interceptés, peuvent révéler les informations traitées par les systèmes informatiques, de communications ou de chiffrement.

TranSec, contient les mesures qui empêchent l'interception des messages transmis non chiffrés. Un des moyens que TranSec met en œuvre pour assurer un niveau de sécurité, est l'utilisation des techniques d'étalement de spectre des signaux avec une logique de changement des valeurs des paramètres au cours de la transmission.

CryptoSec, est la composante de ComSec qui assure la confidentialité de l'information par des algorithmes de chiffrement robustes de point de vue cryptographique et correctement implémentés dans des crypto-systèmes.

1.1.4. Services de sécurité

Pour assurer la sécurité de l'information il faut offrir les services suivants:

- *Confidentialité des données*: réalisée par le chiffrement de l'information. Les données claires sont transformées en données chiffrées et ainsi, elles sont inaccessibles pour les personnes non autorisées.
- *Intégrité*: le destinataire d'un message peut vérifier si le message émis a été modifié involontairement (erreurs de transmission) ou volontairement par un adversaire au cours de sa transmission.
- *Authenticité*: le destinataire d'un message peut vérifier son origine (authentification de la source d'envoie du message).
- *Non-répudiation*: l'émetteur d'un message ne peut pas nier avoir transmis ce message.
- *Disponibilité :* l'information et les services du réseau de communications doivent être disponibles à tout moment.

L'assurance d'InfoSec, est l'implémentation des mécanismes qui assurent un ou plusieurs services de sécurité. C'est la politique de sécurité des opérateurs qui impose, selon les besoins de sécurité et les caractéristiques du réseau en question, le type de services de sécurité à implémenter.

1.1.5. Types d'attaques

Un système de communications sécurisées peut être soumis à deux types d'attaques: les "attaques techniques" et les "attaques pratiques". Une attaque technique essaie de trouver l'information utile à partir du message chiffré ou de la clé secrète, avec des outils mathématiques. A cet effet, soit on analyse l'algorithme de chiffrement/déchiffrement, soit on analyse les messages clairs et chiffrés, soit on peut essayer des clés secrètes. Ces types d'attaques ont eu de succès. Par exemple l'algorithme DES (Data Encryption Standard) qui était considéré comme un algorithme très robuste dans les années 70 est maintenant considéré très faible.

Des résultats beaucoup plus spectaculaires, ont été obtenus par des "attaques pratiques" qui visent à obtenir l'information secrète par: vol, chantage, pot-de-vin, séduction, etc. Il est nettement

moins cher de corrompre une personne qui a accès à des données secrètes que de concevoir une machine ou un algorithme qui casse le système de chiffrement. Nous pouvons trouver beaucoup d'exemples dans la littérature (Schneier B., 2001), des personnes importantes qui ont été corrompues (le directeur du contre-espionnage de la CIA s'est vendu pour moins de 2 millions de dollars) ou séduites (les marins qui gardaient l'ambassade américaine à Moscou).

1.2. Terminologie et concepts de base

Dans la littérature spécialisée, on peut trouver des définitions variées des notions utilisées. Pour une meilleure compréhension, nous allons expliquer les notions utilisées dans cette thèse.

1.2.1. Domaine du chiffrement

- La lettre *A,* signifie un ensemble fini, appelé *alphabet de définition*. Par exemple, *A={0, 1}* est l'alphabet binaire largement utilisé comme alphabet de définition. Notons que tous les alphabets peuvent être exprimés en termes de l'alphabet binaire. Par exemple, chaque lettre de l'alphabet anglais peut être assignée à un mot unique de cinq caractères binaires.
- La lettre *M,* indique un ensemble appelé *l'espace réservé au message*. *M* est composé des séquences de symboles qui appartiennent à l'alphabet de définition. Un élément du *M* est appelé *texte clair*. *M*, peut être composé des séquences binaires, texte alphabétique, code informatique, etc.
- La lettre *C,* indique un ensemble appelé *l'espace réservé au texte chiffré*. *C* est composé des séquences de symboles qui appartiennent à un alphabet de définition qui peut être différent de l'alphabet de définition de l'espace réservé au message. Un élément de *C* est appelé *texte chiffré.*

1.2.2. Transformations cryptographique

- La lettre *K,* indique un ensemble appelé *espace des clés*. Un élément de *K* est appelé une *clé*.
- Chaque élément $e \in K$, détermine une bijection unique de *M* à *C*, notée par E_e et appelée *fonction de chiffrement*. Il est très important que E_e soit une fonction bijective. En effet, ceci garantit que le processus de chiffrement puisse être inversé afin que le texte clair puisse être récupéré pour chaque texte chiffré.
- Chaque élément $d \in K$, détermine une bijection unique de *C* à *M*, notée par D_d et appelée *fonction de déchiffrement*. D_d est une fonction inverse de la fonction E_e.
- Le processus qui applique la fonction E_e pour un message $m \in M$ est appelé *chiffrement de m*.

- Le processus qui applique la fonction D_d pour un message $c \in C$ est appelé *déchiffrement de c*.
- Un *algorithme de chiffrement* contient un jeu $\{E_e : e \in K\}$ de transformations de chiffrement et un jeu correspondant $\{D_d : d \in K\}$ de transformations de déchiffrement avec la propriété que pour chaque $e \in K$ il y a un $d \in K$ unique tel que $D_d = E_e^{-1}$. C'est-à-dire $D_d[E_e(m)]=m$ pour tout $m \in M$.
- Les clés *e* et *d*, sont appelées *paire des clés* ou clés cryptographiques. Pour les systèmes de chiffrement/déchiffrement symétriques, les clés *e* et *d* sont identiques.

1.2.3. Entités communicantes

- Une *entité* est une personne, organisation, dispositif ou processus qui envoie, reçoit ou manipule l'information.
- Un *expéditeur* est une entité légitime qui envoie l'information. Dans la figure 1.2, Alice est l'expéditeur.
- Un *destinataire* est une entité légitime qui reçoit l'information. Dans la figure 1.2, Bob est le destinataire.
- Un *adversaire* ou *attaquant* est une entité qui essaie de contrecarrer les mesures de sécurité. Les actions de l'adversaire peuvent être très variées en fonction de ses intentions et du système de communications. L'attaquant essaye par exemple de se faire passer pour le destinataire ou pour l'expéditeur d'un message. Dans la figure 1.2, Eve est l'adversaire.

1.2.4. Canaux de communications

- Un *canal* de communication est un media de transmission de l'information d'une entité à une autre.
- Un *canal sécurisé physiquement,* est un canal qui n'est pas physiquement accessible à l'adversaire.
- Un *canal public sécurisé,* est un canal qui n'est pas normalement accessible à l'adversaire par des moyens cryptographiques.

Ces éléments sont montrés dans la figure 1.2. Nous montrons l'exemple d'un système de communication entre deux entités, Alice et Bob, qui veulent avoir le service de confidentialité. Ils vont commencer par établir la paire des clés *(e, d)*. Si Alice veut envoyer un message $m \in M$ à Bob elle effectue $c = E_e(m)$ et l'envoie à Bob. Apres avoir reçu *c*, Bob effectue $D_d(c) = m$ et ainsi recouvre le message original *m*. Eve est l'adversaire. Elle peut intercepter le message *c*, et peut utiliser des techniques de cryptanalyse pour trouver le message secret *m*.

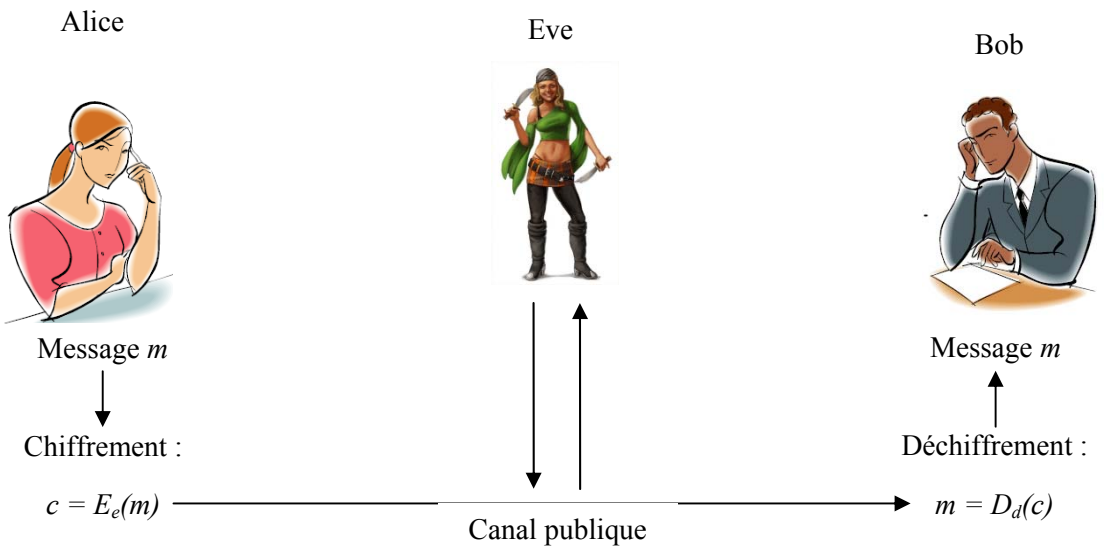


Figure 1.2 : Modèle de canal de communication sécurisé

1.2.5. Cryptologie

- La c*ryptologie* est l'étude de la cryptographie et de la cryptanalyse.
- La *cryptanalyse* est la science des techniques mathématiques qui ont pour but de contrecarrer les techniques cryptographiques.
- Un *cryptanaliste* est une personne qui pratique la cryptanalyse.
- Un *crypto-système* est un terme général qui désigne un ensemble de primitives cryptographiques utilisées pour offrir des services InfoSec.

1.2.6. Autres définitions

- *Protocole*: un algorithme, qui décrit les étapes que chaque entité doit suivre pour atteindre un certain objectif.
- *Préambule (Timestamp)*: une séquence de caractères, qui contient l'information nécessaire pour situer un évènement dans le temps. Dans la plupart de cas, il s'agit d'une date et d'une heure.
- *Nonce cryptographique*: un nombre, utilisé seulement une fois dans un protocole cryptographique.
- *Matériel pour les clés*: des données requises pour établir et maintenir la synchronisation des clés cryptographiques.
- *Signature numérique*: résultat d'une transformation cryptographique des données, afin d'apporter les services suivants: authentification, intégrité des données, non-désaveu des signataires.
- *Mot de passe*: mot composé de lettres, de nombres et d'autres symboles, utilisé pour vérifier une identité ou une autorisation d'accès.

- *Compromission*: divulgation, modification, ou substitution des données sensibles.
- *Cryptopériode*: période de temps pendant laquelle une clé peut être est utilisée.

1.3. Sécurité des données

Dans la conception d'un crypto-système, il faut tenir compte du concept fondamental, qui stipule que tous les jeux *M, C, K,* $\{E_e : e \in K\}$ $\{D_d : d \in K\}$ sont des données publiques. Quand deux entités veulent communiquer de manière sécurisée, le seul secret est la paire des clés *(e, d)* utilisée. Certes, elles peuvent obtenir plus de sécurité si elles gardent en secret leurs algorithmes, mais cette mesure est auxiliaire. L'histoire a montré que garder secret les algorithmes est non efficace, car tôt ou tard ce secret sera divulgué d'une manière ou d'une autre.

1.3.1. Chiffrement à clé symétrique

Considérons le processus de chiffrement/déchiffrement qui contient les transformations $\{E_e : e \in K\}$ et $\{D_d : d \in K\}$ selon la description faite dans le paragraphe 1.2.2. Ce processus est dit à clé symétrique, si pour chaque paire de clés *{e, d}* il est très facile de déterminer la clé *d* à partir de la clé *e* ou inversement. Ce type de processus est appelé à clé symétrique puisque dans la plupart des cas $e=d=k$. Autres appellations utilisées sont : algorithme à clé privée ou chiffrement conventionnel.

La figure 1.3, montre une communication entre deux entités utilisant un algorithme de chiffrement/déchiffrement à clé symétrique.

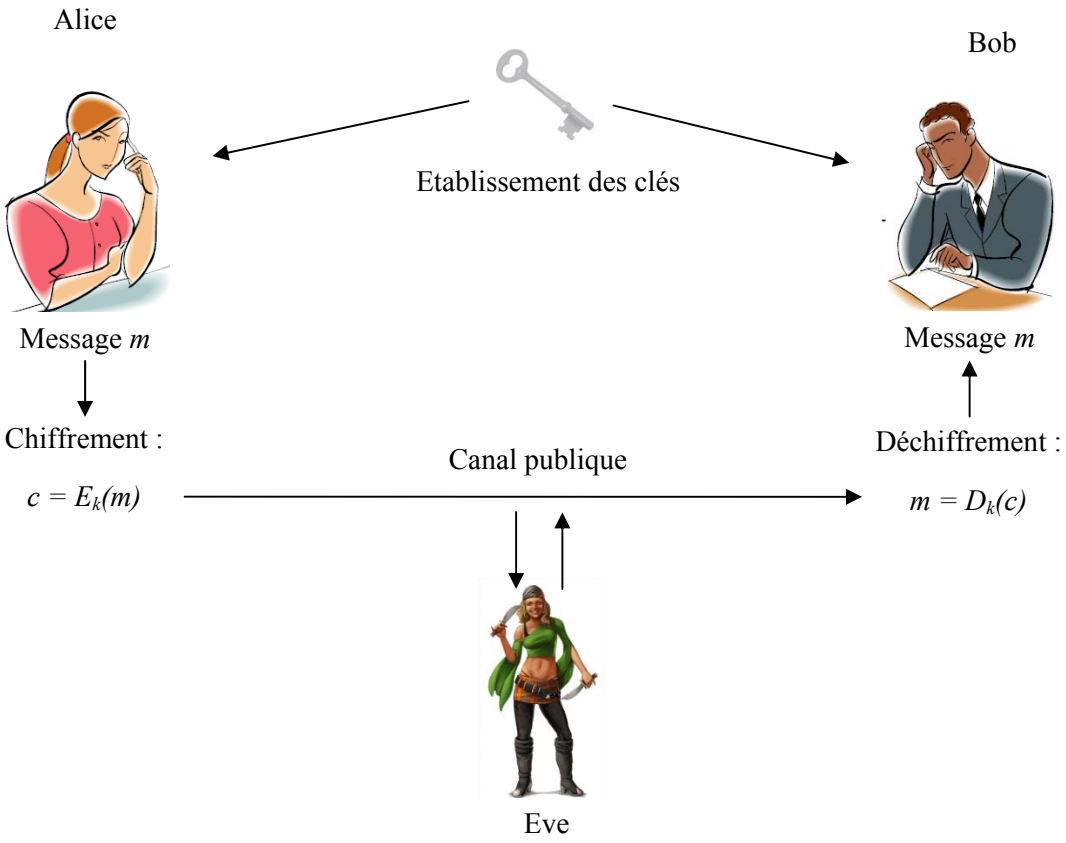


Figure 1.3: Processus de chiffrement/déchiffrement à clé symétrique

Un des défis le plus important pour les systèmes de communications protégés, basés sur des algorithmes de chiffrement/déchiffrement à clé symétrique, est d'instaurer un protocole approprié de gestion des clés.

1.3.1.1. Algorithmes de chiffrement par bloc

Un algorithme de chiffrement par bloc, décompose le texte clair en blocs de taille généralement fixe et chiffre les blocs les uns après les autres. Les plus connus des algorithmes de chiffrement à clé symétrique sont des algorithmes de chiffrement par bloc tels que: l'AES, le DES, le Blowfish, et l'ECKBA modifié, etc.

1.3.1.2. Algorithme de chiffrement par flux

Les algorithmes de chiffrement par flux, traitent les données symbole par symbole. Dans la plupart des cas, il s'agit d'un générateur de nombres pseudo-aléatoires, avec lequel une opération simple est effectuée entre sa sortie et le texte clair. L'opération effectuée est généralement un XOR ou une addition modulo N.

Les algorithmes de chiffrement par flux sont considérés moins robustes, mais plus rapides que les algorithmes de chiffrement par bloc.

1.3.2. Chiffrement à clé publique

Considérons l'algorithme cryptographique qui contient les transformations $\{E_e : e \in K\}$ et $\{D_d : d \in K\}$. Cet algorithme est dit algorithme à clé public, si pour chaque paire des clés $\{e, d\}$ la clé e est disponible au public et l'autre d, est secrète. Pour que tel algorithme soit sûr, il doit être techniquement infaisable de calculer la clé d à partir de la clé e.

La figure 1.4, montre une communication entre 2 entités, utilisant un processus de chiffrement/déchiffrement à clé publique.

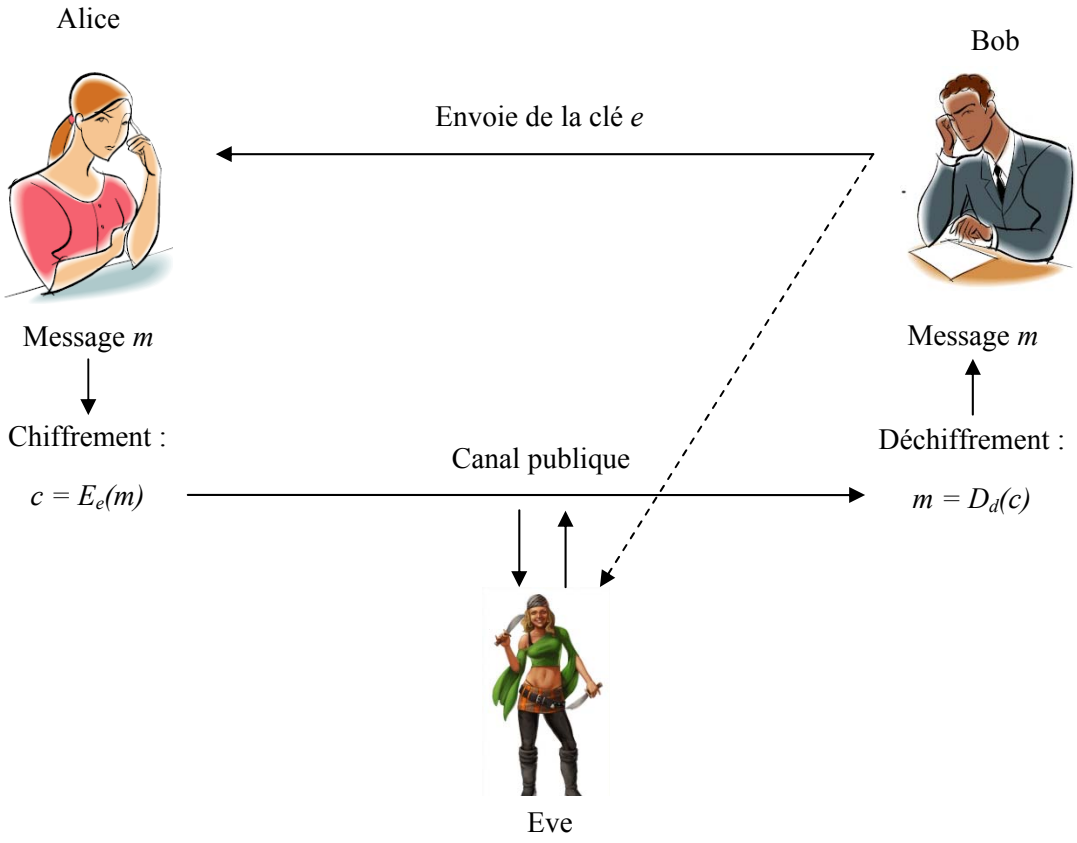


Figure 1.4: Processus de chiffrement/déchiffrement à clé publique

Dans cette figure, Alice veut envoyer à Bob un message sécurisé avec un algorithme à clé publique. Au début, Bob choisi une paire des clés $\{e, d\}$ et envoie la clé e à Alice. Cette clé peut être interceptée par l'adversaire Eve. Ensuite, Alice chiffre le message m avec la clé e et l'envoie à Bob. Quand Bob reçoit le message il utilise sa clé secrète, d, pour déchiffrer le message.

Nous pouvons faire une analogie physique avec un coffre ayant un verrou à combinaison. Dans ce scénario, Bob est le seul qui connait la combinaison qui ouvre le coffre. Il laisse le coffre ouvert et l'envoie à Alice. Alice met un message dans le coffre, puis, le ferme et l'envoie à Bob, qui est le seul capable d'ouvrir le coffre et avoir accès au message.

1.4. Chaos

Une fonction est dite chaotique, si elle est non linaire, ou avec très peu de linéarité, et surtout si elle est sensible aux modifications, mêmes extrêmement faibles de la valeur de la clé secrète, formée par les conditions initiales et les paramètres du système. En effet, si deux systèmes chaotiques identiques ont des états initiaux ou des paramètres qui diffèrent de très peu, les orbites chaotiques de ces systèmes seront très différentes. Cette caractéristique de l'hyper sensibilité à la clé secrète, est à l'origine de nombreux travaux de recherche scientifique, montrant l'apport des signaux chaotiques dans la sécurité des systèmes de communications.

1.4.1. Chaos et cryptographie

Nous avons déjà mentionné, que certaines des propriétés des fonctions chaotiques sont similaires à des propriétés que nous trouvons dans les systèmes cryptographiques. Ce qui a conduit au développement de nouveaux type des crypto-systèmes, les crypto-systèmes basé chaos. Ces propriétés sont présentées dans le tableau 1.1.

Tableau 1.1. Correspondance entre les requis cryptographiques et les propriétés chaotiques

Requis cryptographiques	**Propriétés des fonctions chaotiques**
1. *sensibilité aux clés* : un changement d'un bit de la clé génère un texte chiffré totalement différent, pour le chiffrement d'un texte clair identique.	1. *sensibilité aux paramètres* : une petite variation des paramètres du système génère deux trajectoires chaotiques très différentes même si elles partent de la même condition initiale.
2. *sensibilité au texte clair* : un changement d'un bit de texte clair change totalement le texte chiffré, même si la même clé est utilisée.	2. *sensibilité aux conditions initiales* : deux systèmes chaotiques qui partent des conditions initiales qui différent de très peu auront des trajectoires très différentes.
3. *texte chiffré aléatoire* : le texte chiffré doit avoir un fort caractère aléatoire.	3. *ergodicité* : les trajectoires qui partent des points arbitraires ont une distribution uniforme.

La plupart des algorithmes de chiffrement/déchiffrement basés chaos développés dans la littérature, sont des algorithmes à clé symétrique pour le chiffrement/déchiffrement par bloc ou par flux.

1.4.2. Fonctions chaotiques numériques

Le chaos a été implémenté avec succès dans des systèmes de communications analogiques suite aux travaux de Pecora et Carroll des années 1990. L'implémentation numérique a posé plus de problèmes à cause de la précision finie de la représentation numérique. Quand les systèmes chaotiques sont représentés avec une précision finie, elles deviennent inévitablement cycliques et leurs fonctions de distribution et de corrélation se détériorent. La période du cycle ainsi obtenu est parfois largement plus petite que le nombre total des états de la représentation finie.

La figure 1.5, montre une orbite typique d'un système chaotique numérique.

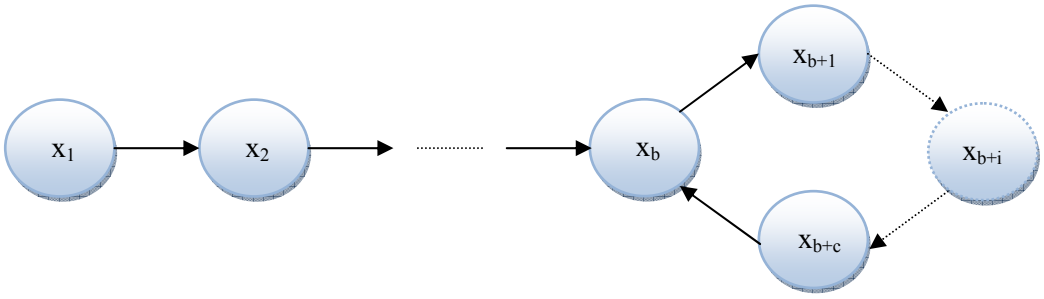


Figure 1.5: Orbite d'un système chaotique numérique

Généralement, l'orbite chaotique contient deux parties:

- *partie transitoire formée de:* $x_1, x_2, ..., x_b$
- *partie cycle formée de* : $x_{b+1}, x_{b+2}, ..., x_{b+c}$

Aussi, *b* est la longueur du régime transitoire et *c* la longueur du cycle. Notons que si *b=0*, l'orbite est un simple cycle et que si *c=0*, l'orbite converge vers un point fixe x_b.

1.4.3. Remèdes

Trois méthodes ont été proposées pour remédier aux problèmes posés par la représentation finie:

- *Mise en cascade de plusieurs systèmes chaotiques*: l'amélioration n'est pas importante parce qu'on obtient une autre fonction chaotique qui aura à son tour des problèmes spécifiques à la représentation finie.
- *Utilisation d'une précision plus élevée* : apporte des améliorations importantes mais, le coût d'implémentation très élevé.
- *Perturbation de l'orbite chaotique* : un générateur de nombres pseudo aléatoires est utilisé pour perturber la sortie du système chaotique numérique.

La figure 1.6, montre le principe de la perturbation :

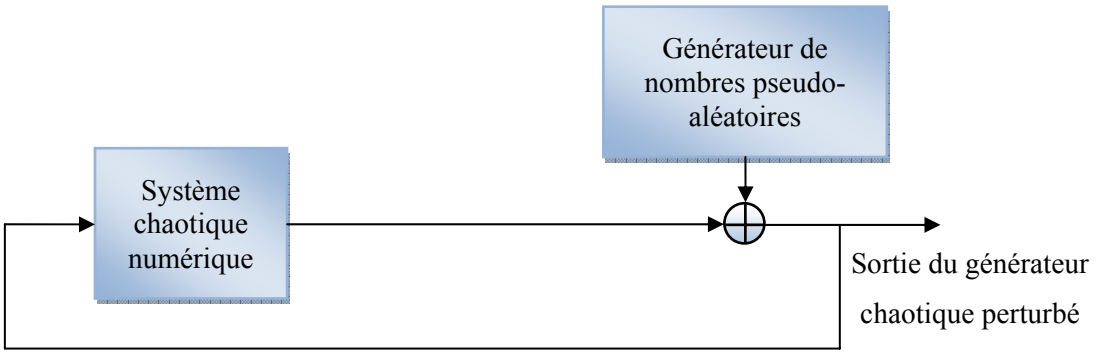


Figure 1.6: Perturbation des systèmes chaotiques numériques

1.5. Gestion de la clé

La gestion de la clé est l'ensemble des techniques et procédures qui ont pour but l'établissement et le maintien d'un état où plusieurs entités communicantes partagent des données communes pour mettre en œuvre des techniques cryptographiques.

Le domaine de gestion des clés est très important pour les deux types d'attaques, technique et pratique, déjà décrits dans le paragraphe 1.1.5.

Pour se protéger contre les attaques techniques, la gestion des clés impose les requis suivants:

- *la durée de vie des clés secrètes* doit être telle qu'un adversaire n'ait pas assez de textes en clair et chiffré avec la même clé pour l'empêcher de déterminer la clé;
- *la modalité de génération des clés* doit être telle qu'elle soit parfaitement aléatoire et valide (certaines algorithmes de chiffrement ont de mauvaises résultats avec certaines clés),
- *les protocoles de transfère des clés* doivent être extrêmement robustes.

En ce qui concerne les attaques pratiques, la gestion des clés a pour but de réduire le nombre de points sensibles (personnes, serveurs, etc.) et de les identifier correctement. Nous présentons quelques techniques de gestion des clés dans l'annexe A.3.

Un algorithme cryptographique est dit cassable, si un adversaire peut défaire les services de sécurité offerts par l'algorithme (confidentialité, authentification, intégrité, etc.) de manière systématique, sans avoir connaissance de la paire des clés (e, d) et dans un temps réaliste.

Une période de temps convenable implique le fait que la protection des données est encore utile. Par exemple, l'instruction d'acheter des actions en Bourse, doit être tenue secrète seulement pour quelques minutes, tandis que certains secrets d'état doivent être tenus secrets pour une période de temps indéfinie.

Un algorithme cryptographique peut être cassé, si toutes les paires des clés (e, d) sont vérifiées. Ce type d'attaque est dit, attaque par force brute ou attaque exhaustive. Par conséquence, le nombre des clés doit être assez important pour que cette attaque ne soit pas techniquement faisable.

Généralement on utilise pour l'espace des clés, l'alphabet de définition binaire. Pour cela les clés sont exprimées en bits. L'attaque par force brute impose une longueur minimale pour les clés. Selon les recommandations de NIST (Barker E., 2006), la taille minimale de la clé symétrique pour une bonne sécurité des données est de 128 bits.

Un autre aspect qui doit être pris en compte quand on analyse la taille de la clé, est l'entropie de la clé. Certains algorithmes qui ont une clé de taille 128 bits, mais seulement une partie des combinaisons peuvent être utilisées. Dans ce cas, la taille de la clé, de point de vue cryptographique, doit être donnée par son entropie.

1.6. Protection des données Multimédia

Nous assistons à la migration des documents du format papier vers le format numérique. Ceci révèle des problèmes relatifs au remplacement des sceaux, timbres et signatures analogiques par leurs versions numériques. Quand il s'agit de l'information numérique, c'est très difficile de faire une distinction claire entre l'original et les copies, ou de dépister les modifications malveillantes subies par les données informatives. La voie est ouverte pour la violation du droit d'auteur ou pour les modifications malveillantes.

Le tatouage numérique est une des solutions pour résoudre ces problèmes. Il permet le transfert et l'utilisation sécurisés des images. Le tatouage numérique consiste à insérer un message porteur d'information dans un média-hôte (une image, un signal audio ou un signal vidéo), et permet une modalité de vérification de l'intégrité des données ou un mécanisme de revendication des droits d'auteur.

1.6.1. Structure générale de la technique de tatouage numérique

La structure de base du tatouage numérique contient trois parties (voir figure 1.7)

- *Insertion du tatouage numérique* qui doit respecter les conditions de transparence et de robustesse;
- *Attaques*;
- *Détection et extraction*.

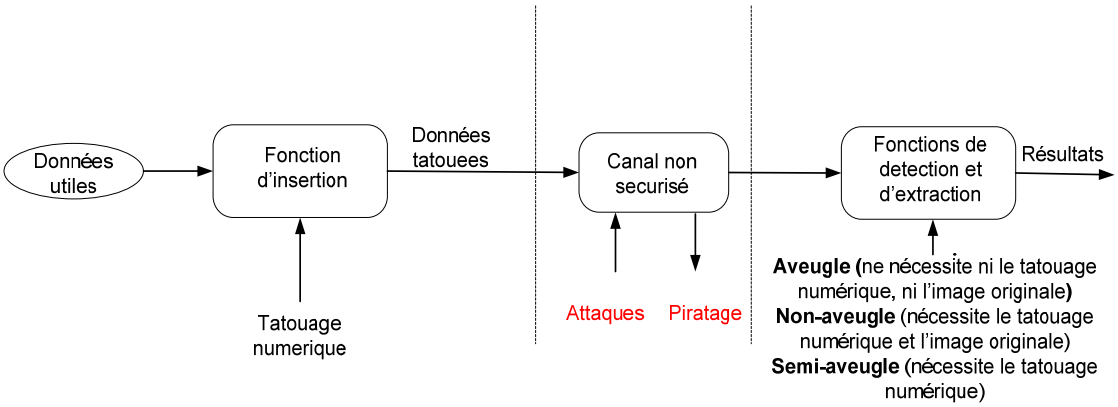


Figure 1.7. Structure générale de la technique du tatouage numérique

1.6.2. Classification des techniques de tatouage numérique

Les techniques de tatouage numérique peuvent être classifiées en fonction de plusieurs critères (Mohanty, 1999). En effet, d'après le domaine de travail, la classification se fait selon les domaines spatial ou fréquentiel. La classification peut se faire aussi selon le type de média hôte, à savoir : texte, image, fichier audio ou vidéo. Aussi, en fonction de la perception humaine, les algorithmes de tatouage numérique peuvent être divisés en visibles et invisibles. Les algorithmes invisibles se répartissent en deux catégories: les algorithmes robustes qui doivent assurer la protection des droits d'auteur du propriétaire et les algorithmes fragiles qui sont utilisés pour tester l'intégrité des données numériques. D'après la modalité d'extraction, les méthodes robustes de tatouage numérique peuvent être classifiées en aveugles, semi-aveugles ou non-aveugles. Pour les méthodes aveugles, le processus d'extraction n'a besoin ni de l'image originale, ni du tatouage inséré. Les méthodes semi-aveugles exigent seulement la connaissance du tatouage inséré et dans le cas des méthodes non-aveugles, le processus nécessite la connaissance de l'image originale et aussi du tatouage inséré.

Enfin, on peut aussi classifier les méthodes de tatouage numérique comme méthodes standard ou méthodes basées sur les signaux chaotiques.

La figure 1.8 montre le diagramme de classification des méthodes de tatouage.

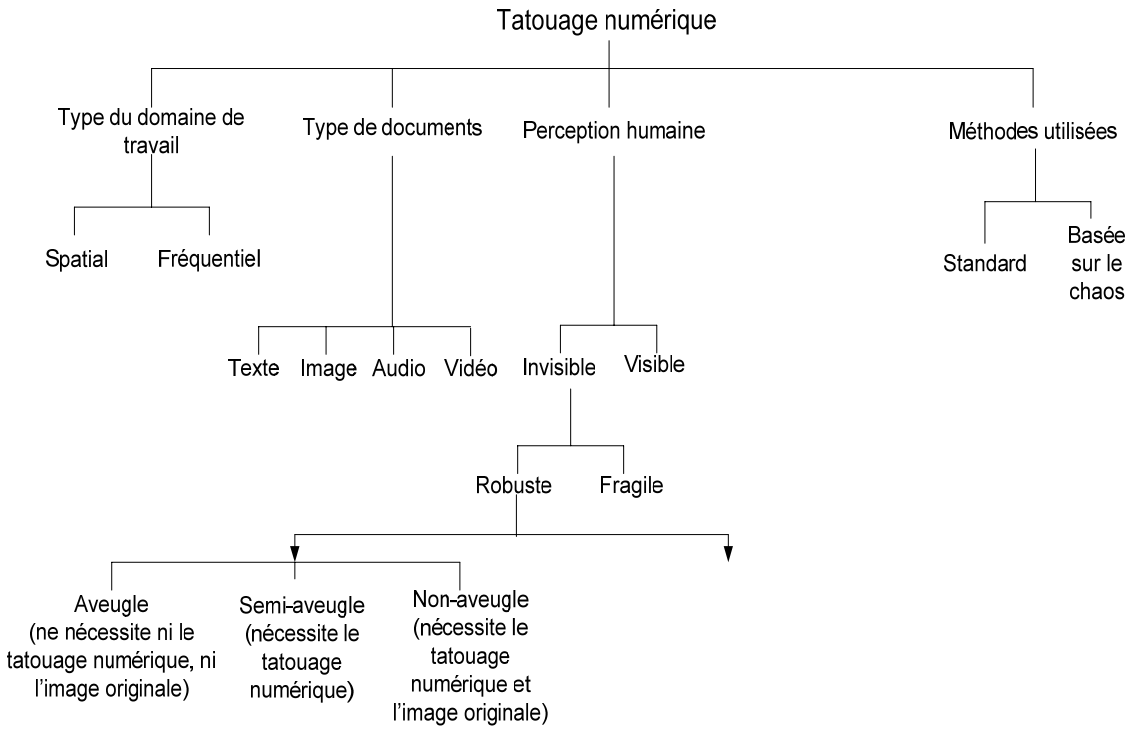


Figure.1.8. Classification des techniques du tatouage numérique

Le tatouage fragile assure le service d'intégrité des données, tandis que le tatouage robuste est utilisé pour renforcer les droits d'auteur.

2. Communications IP par DVB-S sécurisées par des séquences chaotiques

2. Communications IP par DVB-S sécurisées par des séquences chaotiques

2.1. Généralités

La technologie de communications de nos jours est caractérisée par une tendance très puissante vers le « tout IP » (Internet Protocol). L'Internet, réseau basé IP, s'est imposé comme le plus important moyen de communication du point de vue du nombre d'utilisateurs (plus de 25% de la population du monde, selon le site internetworldstats.com, (2010)) et aussi du point de vue des services offerts (exemple : lire l'actualité, faire des achats, gérer des comptes bancaires, etc).

Il existe actuellement une puissante dorsale Internet, qui offre des services à haut débit pour la plupart des utilisateurs. Cependant, dans beaucoup de cas, le problème du « dernier kilomètre» reste à être résolu. Une des solutions pour ce problème est le DVB (Digital Video Broadcasting) Satellitaire: DVB-S, DVB-S2, et DVB-RCS (Return Channel via Satellite).

L'Internet par satellite présente beaucoup de contraintes, en termes de vitesse et temps de réponse, mais il est pratiquement la seule solution viable dans les cas où les connexions IP classiques ne sont pas possibles, ou ne sont pas désirées. Dans le cas où les connexions classiques à Internet ne peuvent pas être une alternative, nous pouvons citer deux scénarios d'utilisation des communications IP par satellite: le premier scénario est le cas des endroits isolés: les sommets des montagnes, les plateformes pétrolières, les villages isolés, les déserts, etc. Le second scénario est le cas des engins mobiles à grande vitesse tels que: les avions, les navires, les trains à grand vitesse, etc.

Dans le cas où les connexions classiques à Internet ne sont pas souhaitées, nous citions le cas des entreprises ou organisations implantées mondialement qui, pour des raisons de sécurité et de tranquillité, ne veulent dépendre que d'un seul opérateur et non de plusieurs opérateurs différents appartenant à des pays différents.

Un des défauts majeurs des communications IP par DVB satellitaire est le faible degré de la sécurité de ces communications. Ceci est dû, d'une part aux caractéristiques intrinsèques des communications satellitaires qui font que le message émis par le satellite est disponible à un nombre illimité des récepteurs et, d'autre part, au manque d'une solution de sécurité performante pour ces communications.

Nous présentons dans ce chapitre les communications IP par DVB via Satellite et nous proposons une solution de sécurité plus performante que les solutions actuelles. Cette solution s'appuie d'une part, sur l'apport des séquences chaotiques, et d'autre part, sur une gestion de clés multicouches. Notre système de sécurité répond aux besoins de sécurité des communications IP par satellite et tient compte des caractéristiques des communications satellitaires.

Dans la première partie du chapitre, nous présentons le contexte de notre étude: l'importance et l'état de l'art des communications IP par satellite. Nous abordons les besoins de sécurité de ces communications et les solutions existantes à ce moment. Puis, nous proposons une solution originale de sécurité pour les communications IP par DVB-S avec encapsulation ULE (Unidirectional Lightweight Encapsulation). Ensuite, nous analysons les performances de cette solution en termes de taux des données ajoutées et nous montrons que les résultats obtenus sont meilleurs aux solutions de sécurité existantes actuellement. Enfin, nous concluons ce chapitre en montrant que la solution de sécurité proposée peut être utilisée, non seulement pour les cas des communications unicast, mais aussi dans les cas des communications multicast.

2.2 Famille des standards DVB

2.2.1. Projet DVB

Le projet DVB a été créé en 1991 suite à une collaboration de plusieurs compagnies pour concevoir des technologies de diffusion de télévision numérique par satellite, par câble et par antenne radio. Plus précisément, ce projet est né d'un consortium européen d'environ 300 compagnies, de plus de 35 pays, du domaine de télécommunication (producteurs d'équipements, chaines de télévision, institutions de standardisation). La famille des standards DVB a la caractéristique d'être ouverte et interopérable pour différents types de communications. Les concurrents sont la norme ATSC (Advanced Television System Committee) utilisée aux Etats Unis et la norme ISDB (Integrated Services Digital Broadcasting) utilisée au Canada, au Japon et en Amérique du Sud.

Le projet DVB, se compose de deux modules : le premier est le Module Commercial (CM – Commercial Module) chargé d'établir un ensemble « d'Exigences Commerciales » (Commercial Requirements) qui tiennent compte des besoins du marché sans se préoccuper des aspects techniques. Le second est le Module Technique (TM – Technical Module) qui précise les spécifications techniques à réaliser tout en respectant les Exigences Commerciales. Le projet est divisé en plusieurs tâches, dont chacune est analysée par des équipes appartenant au CM et au TM. Une fois qu'un brouillon des spécifications techniques est rédigé par le TM et le CM, il est envoyé au « DVB Project Steering Board » pour l'approbation finale avant d'être soumis à un organisme international de normalisation, normalement l'ETSI (European Telecommunications Standard Institute).

2.2.2. Famille des standards DVB satellitaire

La famille des standards DVB a été conçue, à l'origine, pour permettre l'émission de programmes de télévision avec interopérabilité entre les systèmes de communication satellitaires: DVB-S, DVB-S2, DVB-RCS (DVB – Return Channel via Satellite), terrestres : DVB-T (DVB – Terrestrial), câblés DVB-C (DVB-Cable) et mobiles : DVB-H (DVB – Handheld). Chaque standard a

son propre code correcteur d'erreurs FEC (Forward Error Correction) et technique de modulation, mais ils utilisent tous le même multiplexage des données : le MPEG-2 (Moving Pictures Expert Group) pour la première génération, MPEG-2, MPEG-4 ou le GSE (Generic Stream Encapsulation) pour la deuxième génération.

Ensuite, les standards DVB sont devenus porteurs pour d'autres types de protocoles de communication, tel que l'IP, pour lequel le DVB assure la couche physique et la couche liaison des données.

Le premier standard DVB, proposé en 1993 pour la transmission satellitaire était le DVB-S. Il utilise la modulation QPSK, le codage Reed-Solomon (204,188) et un code FEC (Forward Error Correcting) (ETSI, EN 300.421). La première application commerciale a été implémentée en 1995 par l'opérateur de télévision français, Canal Plus.

Le standard DVB-S2 a été proposé dix ans plus tard, en 2003, comme la nouvelle génération de DVB-S. Il propose l'utilisation des avancées technologiques de l'époque, en codage et modulation. Plus précisément le DVB-S2 utilise les modulations: 8PSK, MAPSK, 16APSK, et 32APSK et les technologies VCM (Variable Coding Modulation) et ACM (Adaptive Coding and Modulation) qui permettent le changement des paramètres du codage en temps réel. Ceci augmente l'efficacité de l'utilisation de la ressource spectrale et permet ainsi, des débits plus élevés pour la même bande de fréquence.

Actuellement, et en dépit des avantages évidents du DVB-S2 par rapport au DVB-S, le nombre des systèmes qui utilisent le DVB-S est approximativement égal au nombre des systèmes qui utilisent le DVB-S2 et le projet DVB ne prévoit pas dans le futur proche, une migration complète du DVB-S vers le DVB-S2. Ceci vient du fait de l'existence déjà de millions des terminaux DVB-S qui fonctionnent très bien et aussi parce que la migration vers la technologie DVB-S2 impliquerait des coûts supplémentaires.

Le DVB-S et le DVB-S2 sont utilisés pour la voie descendante. Le DVB-RCS proposé en 1999, est utilisé pour la voie montante, puisqu'il définit les caractéristiques du canal de réponse des terminaux.

Tous les terminaux partagent les mêmes fréquences et utilisent la technique de multiplexage MF-TDMA (Multi-Frequency Time Division Multiple Access) qui implique la synchronisation des terminaux.

2.3. Communications IP via DVB-S

2.3.1. Architecture du système

En général, il y a deux architectures du DVB satellitaire pour transporter des données IP: la première (voir figure 2.1 a) concerne les réseaux utilisés par les fournisseurs des services IP, pour

offrir des services à leurs clients (accès Internet, VoIP (Voice over IP), conférences, etc.). La seconde (voir figure 2.1 b) offre une liaison, par satellite, entre deux grands réseaux (réseaux métropolitains, régionales et nationales).

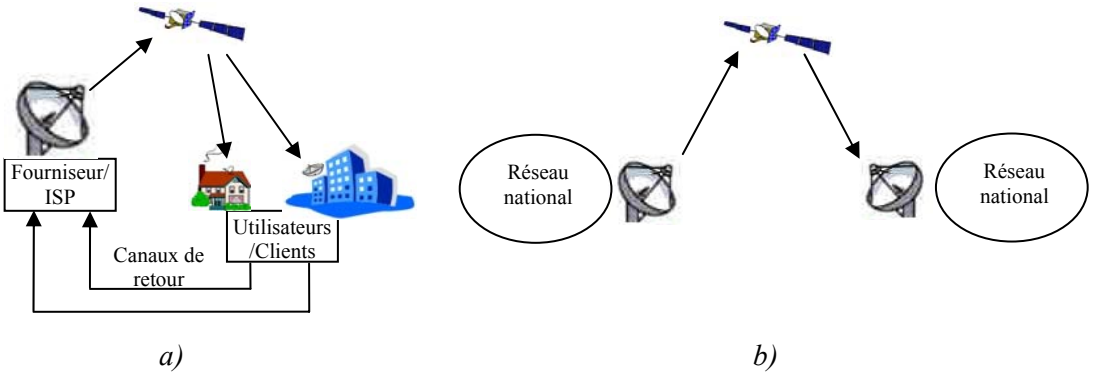


Figure 2.1: Architecture des systèmes de communications IP par DVB-S, (a) liaison fournisseur IP-clients, (b) liaison entre deux grands réseaux

Pour le premier cas, le fournisseur, qui est en général un ISP (Internet Service Provider) a une connexion à Internet et transmet les données d'Internet à ses clients. Le réseau a une topologie en étoile, avec le fournisseur au centre. Contrairement à la diffusion de la télévision, les communications IP nécessitent, dans la plupart des cas, un canal de retour, qui permet aux utilisateurs d'envoyer des messages vers le fournisseur. Cette connexion peut être une liaison par ligne commutée, GPRS (General packet radio service), satellitaire, etc. Une des caractéristiques la plus importante des communications IP est la forte asymétrie entre les deux canaux directe et de retour. Pour la plupart des applications (navigation Internet, e-mail), le canal descendant (downlink) est de bande beaucoup plus large que celui du canal montant (uplink). Pour cela, les systèmes qui offrent un accès à Internet par satellites utilisent généralement des canaux différents pour les deux sens de communication. L'accès à Internet par DVB satellitaire n'impose aucune condition pour le canal de retour. En fonction des possibilités et des nécessités des connections, différents types de connexions peuvent être utilisés pour le canal de retour, tels que : Dial Up, ISDN, GPRS, DVB-RCS, et d'autres systèmes de communications satellitaires.

Pour la deuxième architecture, la liaison par satellite entre deux grands réseaux est de type point à point. Dans la plupart des cas, deux connexions sont utilisées, une pour la direction montante et une pour la direction descendante.

Même si la topologie des réseaux et les équipements utilisés sont différents pour les deux architectures de systèmes de communication, les protocoles de communications et les besoins de sécurité sont identiques. En plus, le deuxième système peut être considéré comme un cas particulier du premier système: un fournisseur avec seul utilisateur. Pour cela, dans notre étude, nous nous intéressons à la problématique des protocoles de communications et la sécurité de la première architecture seulement.

2.3.2. Structure du fournisseur et des usagers

2.3.2.1. Structure générale

L'architecture d'une station de base qui permet l'accès Internet par satellite, doit contenir un router avec accès à Internet, un encapsulateur IP, un multiplexeur MPEG-2 un modulateur et une antenne satellite, comme c'est montré dans la figure 2.2.

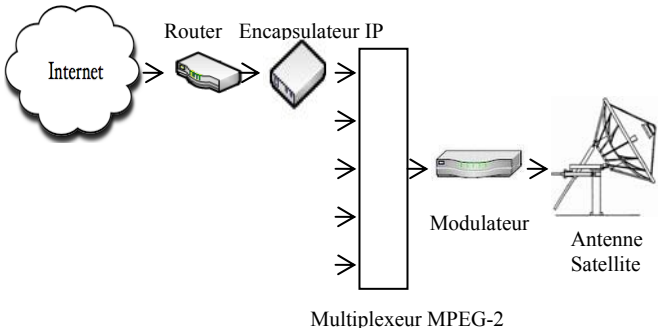


Figure 2.2: Structure du fournisseur

Le router est un équipement IP qui fait la liaison avec l'Internet. L'encapsulateur réalise l'encapsulation ULE, et génère le flux MPEG-2. Sa structure est discutée dans le paragraphe suivant. Le multiplexeur permet de multiplexer les différents flux de données MPEG-2 TS (Transport Stream). Ces flux peuvent contenir des données TV, des données IP, des radios, etc. Le modulateur prépare le flux des données pour pouvoir les transmettre par satellite: il réalise notamment les opérations de codage correcteur d'erreurs, d'entrelacement (interleaving) et de modulation. L'antenne satellite envoie le signal vers le satellite avec la puissance requise et dans la bande des fréquences allouée.

Les stations réceptrices (voir figure 2.3), doivent être équipées avec une antenne satellite, un démodulateur, un démultiplexeur MPEG-2, un récupérateur des paquets IP (IPPRU – IP Packet Recovery Unit), un router et l'utilisateur final qui peut être un ordinateur ou un réseau LAN (Local Area Network).

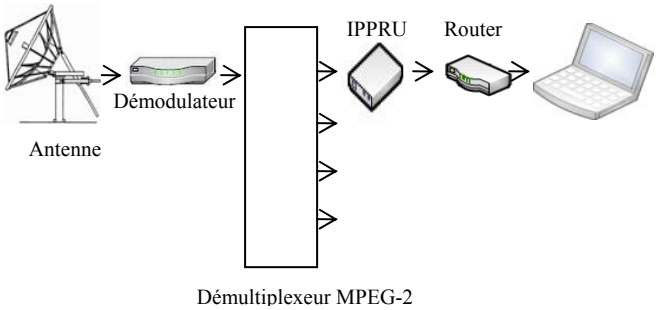


Figure 2.3. : Structure de l'usager

L'antenne reçoit le signal satellitaire et l'envoie au démodulateur qui transforme le signal analogique reçu en un signal numérique et reconstitue le flux MPEG-2-TS. Le démultiplexeur

sélectionne les flux qui sont adressés au destinataire. Le IPPRU réalise l'extraction des paquets IP qui sont envoyés vers l'utilisateur final via un router.

2.3.2.2. Encapsulateur IP et IPPRU

La liaison entre les équipements classiques des réseaux et les équipements de communications DVB-S est réalisée par l'encapsulateur IP. Sa structure et son fonctionnement sont montrés dans la figure 2.4.

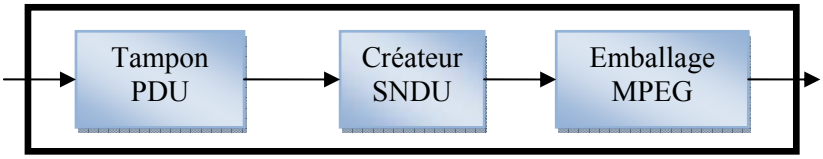


Figure 2.4: Structure de l'encapsulateur IP

Le bloc *Tampon* PDU (Protocol Data Unit) est un bloc qui mémorise les PDU qui arrivent en vue de leur encapsulation.

Le bloc C*réateur* SNDU (SubNetwork Data Unit) ajoute l'en-tête et l'en-queue ULE au PDU. Il traite la longueur de la SNDU, le type du PDU, l'adresse destination, si elle est utilisée, et le code CRC (Cyclic Redundancy Check). Ainsi, les SNDU crées sont prêts à être emballés dans des trames MPEG-2 TS et à être envoyés.

Le bloc *Emballage MPEG* crée les trames MPEG qui contiennent les SNDU reçus.

L'IPPRU utilise une structure montrée dans la figure 2.5, permettant de réaliser les fonctions inverses de l'encapsulateur IP.

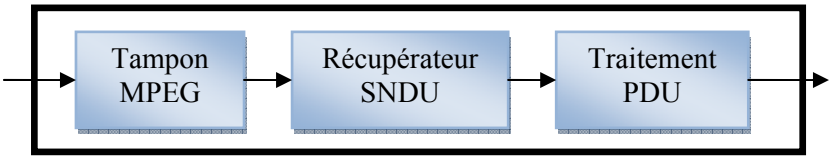


Figure2.5: Structure du IPPRU

Le bloc *Tampon* MPEG mémorise les trames de transport MPEG qui arrivent du démultiplexeur et qui attendent d'être traitées par le récupérateur SNDU.

Le bloc *Récupérateur* SNDU extrait les SNDU contenus dans les trames transport MPEG.

Le bloc *Traitement* PDU extrait les PDU à partir des SNDU reçus et les envoie vers le router.

2.3.3. Transport des paquets IP par DVB satellitaire

Au départ, le standard DVB-S a été conçu pour transporter les émissions de télévision et des données auxiliaires (télétexte, sous titrage, etc). Ultérieurement il est devenu possible de l'utiliser pour permettre l'envoie des paquets IP.

A ce propos, deux types d'encapsulation peuvent être utilisés: l'encapsulation MPE (MultiProtocol Encapsulation) et l'encapsulation ULE (Unidirectional Lightweight Encapsulation).

La figure 2.6, montre le mécanisme de transport des paquets IP par DVB satellitaire avec encapsulation MPE ou ULE.

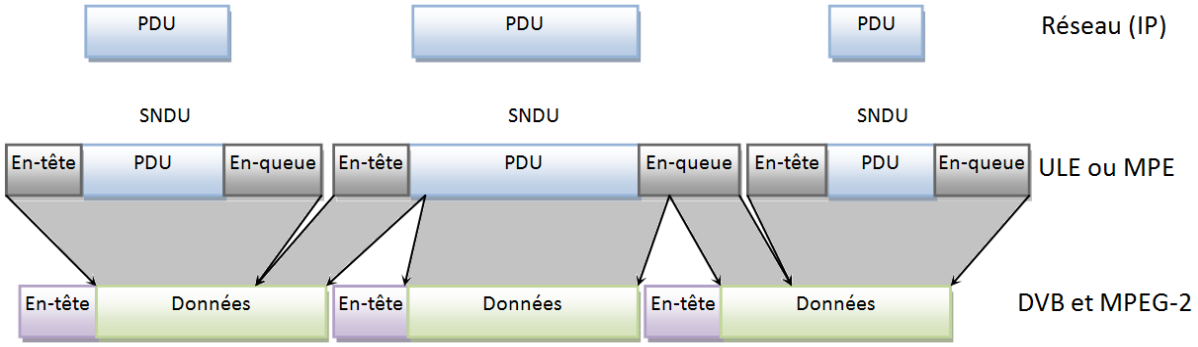


Figure 2.6. Encapsulation ULE ou MPE et transport en trames MPEG-2 des paquets IP

Dans cette figure, nous avons utilisé l'appellation plus générale PDU (Packet Data Unit) pour designer les paquets IP, car les encapsulations présentées peuvent aussi être utilisées pour d'autres protocoles au niveau réseau, et non seulement pour l'IP.

L'encapsulation ajoute à chaque PDU un en-tête et un en-queue pour former les SNDU qui permettent de faciliter le transport des paquets avec des trames MPEG-2 TS. Ensuite, chaque SNDU est transporté avec une ou plusieurs trames MPEG-2 TS de taille fixe égale à 188 octets. A noter que chaque trame MPEG-2 peut transporter au maximum deux SNDU : si après avoir fini l'encapsulation d'un SNDU il reste de l'espace dans la trame MPEG-2, l'encapsulateur peut commencer l'encapsulation d'un nouveau SNDU dans la même trame.

Des études comparatives de deux standards ULE et MPE (G. Fairhurst 2003, Collini-Nocker B. 2004, Hong T.C. 2005) ont montré que l'encapsulation ULE a aux moins trois avantages par rapport à l'encapsulation MPE: support de nouveaux protocoles, meilleure efficacité de transport et coût réduit de traitement. Pour cela, dans notre travaille, nous traitons seulement l'encapsulation ULE.

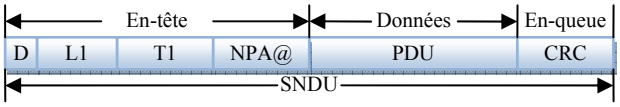
2.3.3.1. Encapsulation ULE

L'en-tête ULE est composé de: 3 champs obligatoires, un champ facultatif et un certain nombre d'extensions d'en-tête. Les champs obligatoires sont:

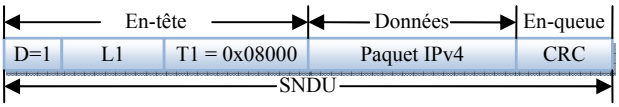
- *Adresse Destination Absent – D :* un bit qui indique la présence ou l'absence du champ facultatif NPA (Network Point of Attachment). Le champ facultatif NPA est représenté sur 6 octets et a une structure et une utilisation similaire avec l'adresse de la couche liaison des données MAC (Media Access Control), utilisée par le protocole Ethernet. Si le champ NPA est absent, la valeur du champ PID (Packet Identifier) de l'en-tête MPEG-2 peut être utilisée comme adresse destination.

- *Longueur – L1 :* un champ sur 15 bits qui indique la longueur de la trame SNDU, plus précisément, le nombre d'octets qui suivent le champ T1, incluant les 4 octets CRC.
- *Type – T1 :* un champ sur 16 bits qui indique: le type du PDU contenu par le SNDU, si sa valeur est supérieure à 0x0600 (1536), ou le type d'extension d'en-tête, si sa valeur est inférieure à 0x0600.

La figure 2.7. (a) montre la structure générale d'un SNDU ULE, et la figure 2.7. (b) montre un exemple d'encapsulation ULE d'un paquet IPv4 comme données.



a)



b) Exemple d'encapsulation ULE pour un paquet IPv4

Figure 2.7: Structure d'un SNDU ULE, (a) structure générale, (b) exemple pour un paquet IPv4 comme données

2.3.3.2. Extension de l'en-tête ULE

L'encapsulation ULE a été conçue dans le but de réduire au minimum le nombre des données supplémentaires ajoutées aux PDU. Pour cela, son en-tête contient seulement 3 champs obligatoires. Mais, si on veut ajouter des services supplémentaires, telle que la sécurité, des informations additionnelles sont nécessaires. Pour cela, l'encapsulation ULE permet l'extension de l'en-tête avec des champs supplémentaires. La structure du type d'extension de l'en-tête est présentée dans la figure 2.8.

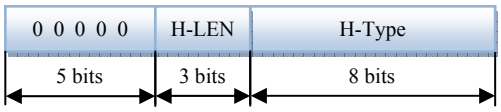


Figure 2.8: Structure du champ T1 pour l'extension de l'en-tête ULE

Les 5 premiers bits du poids fort sont mis à 0 pour signaler qu'il s'agit du type de l'extension de l'en-tête et non du type du PDU transporté. La partie *H-LEN* du champ *T1* indique, si sa valeur est différente de 0, la longueur de l'extension de l'en-tête. Une valeur entre 1 et 5 indique une longueur entre 2 et 10 octets. Si la valeur du champ *H-LEN est* supérieure à 5, le champ sera utilisé avec le champ *H-TYPE* pour indiquer le type *EtherType* (DIX, 1982). Une valeur égale à 0, indique une extension d'en-tête mandataire. Chaque extension d'en-tête mandataire a une longueur prédéfinie qui est connue a priori par le récepteur.

La partie *H-Type* indique le type d'extension de l'en-tête. Il peut être un des 256 extensions mandataires ou un des 256 extensions optionnelles. Les valeurs que ce champ peut prendre sont

décrites par un registre IANA (Internet Assigned Number Authority). Ce registre associe à chaque extension mandataire une longueur prédéfinie.

Le format général de l'en-tête ULE avec une extension, est présenté dans la figure 2.9:

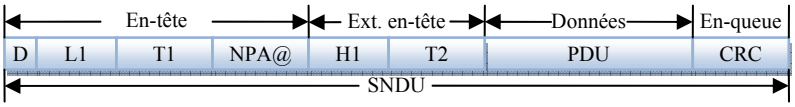


Figure 2.9: SNDU avec une extension de l'en-tête ULE

- *T1* : a une valeur inférieure à 0x0600 spécifiant le type d'extension de l'en-tête.
- *H1* : est composé d'un ensemble des champs qui définissent l'extension de l'en-tête, selon le type d'extension utilisé.
- *T2* : a une utilisation similaire que *T1,* c'est-à-dire, si sa valeur est inférieure à 0x0600 alors, le champ *T2* est suivi par une autre extension d'en-tête et il défini le type de cette extension. Sinon, il est suivi par un PDU et il défini le type du PDU.

Cette manière de structurer l'extension de l'en-tête, permet le chaînage d'un nombre illimité d'extensions de l'en-tête. En effet, si la valeur du champ *Tn* de l'extension de l'en-tête (*n-1*) est plus petite que 0x0600, ceci implique une nouvelle extension appliquée à l'en-tête, l'extension de l'en-tête *n*. La figure 2.10, montre un exemple de SNDU avec 2 extensions successives de l'en-tête:

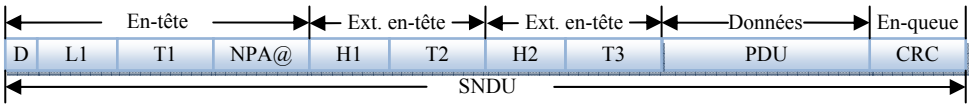


Figure 2.10: SNDU avec deux extensions successives de l'en-tête

2.3.4. Standard MPEG-2 TS

L'encapsulation ULE est développée pour permettre l'envoie des paquets IP via MPEG-2 TS. Le standard MPEG2-TS, permet la transmission ou le stockage: des données, des données vidéo ou des données audio utilisées par le DVB et par d'autres systèmes de communication. Il est utilisé par tous les systèmes DVB-S et par une partie des systèmes DVB-S2, car le DVB-S2 peut utiliser aussi le MPEG-4 ou le GSE, selon le type d'implémentation.

Le format de multiplexage des données MPEG-2 TS, utilise des trames de transport ayant une longueur fixe de 188 octets.

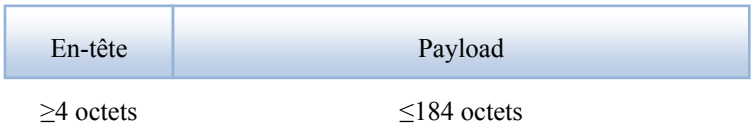


Figure 2.11: Trame de transport MPEG-2

La longueur de 188 octets a été choisie pour permettre l'interopérabilité avec l'ATM (Asynchronous Transfer Mode). Chaque trame MPEG-2 peut être transportée par 4 cellules ATM, chacune de taille fixe égale à 53 octets. Aussi, deux trames MPEG-2 peuvent être transportées par 8 cellules ATM si l'AAL5 (ATM Adaptation Layer 5) est utilisé (d'après le ITU-T Recommendation I.363.5, 1996). Toutes les trames contiennent un en-tête d'au moins de 4 octets. La structure de l'en-tête est présentée dans la figure 2.12.

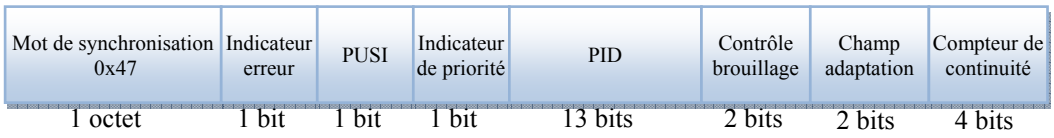


Fig 2.12. En-tête d'une trame MPEG-2

Les champs les plus importants de l'en-tête sont:

- *Mot de synchronisation*: utilisé pour établir la synchronisation du récepteur ou la re-synchronisation en cas d'erreur.
- *Indicateur erreur:* indique s'il y a des erreurs dans la trame MPEG-2 TS.
- *PUSI (Payload Unit Start Indicator):* indique si la trame MPEG-2 contient le début d'un nouveau SNDU, par exemple un nouveau paquet IP encapsulé. Si une nouvelle SNDU commence dans la trame, alors l'en-tête MPEG-2 sera suivi d'un octet qui contient l'adresse du début, le PP (Payload Pointer).
- *PID (Packet Identifier):* indique le flux de données ou programme (télévision) auquel le paquet appartient.

Quand le DVB-S est utilisé pour la diffusion de la télévision, deux types de tableaux, contenant de l'information relative au programme (PSI – Program Specific Information), utilisent le PID:

- Le tableau d'association des programmes (PAT – Program Association Table)
- Le tableau du mappage des programmes (PMT – Program Map Table).

Lors de la transmission de plusieurs programmes de télévision, le PAT permet d'indiquer le programme auquel le TS appartient. Par ailleurs, chaque programme a son propre PMT qui indique le rôle de chaque TS dans le programme: soit un flux vidéo, soit un des flux audio, soit un flux de données (sous-titrage, télétext, etc…). La figure 2.13, montre l'exemple d'une émission de deux programmes TV sur le même flux MPEG2. Le PID du premier programme TV est 20 et celui du deuxième programme TV est 40. Le premier programme TV, contient un flux audio (PID 21), un flux vidéo (PID 22) et un flux avec sous-titrage (PID 23). Le deuxième programme TV, contient un flux audio (PID 41) et un flux vidéo (PID 42). Toutes les trames de transport avec le même PID forment un ES (Elementary Stream).

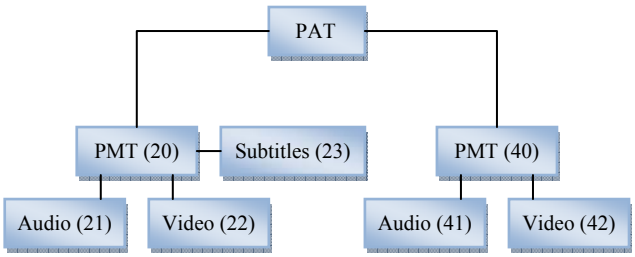


Figure 2.13: Structure du PSI

2.3.5. *Padding* et *packing*

Une fois le SNDU créé, il est prêt à être envoyé dans des trames MPEG-2 TS. L'encapsulation ULE peut être utilisée en deux modes opératoires: *padding* et *packing*.

Dans le cas du *padding*, si une trame MPEG-2 contient la fin d'un SNDU alors le reste de la trame est complété avec 0xFF. Cette procédure est montrée dans la figure 2.14.

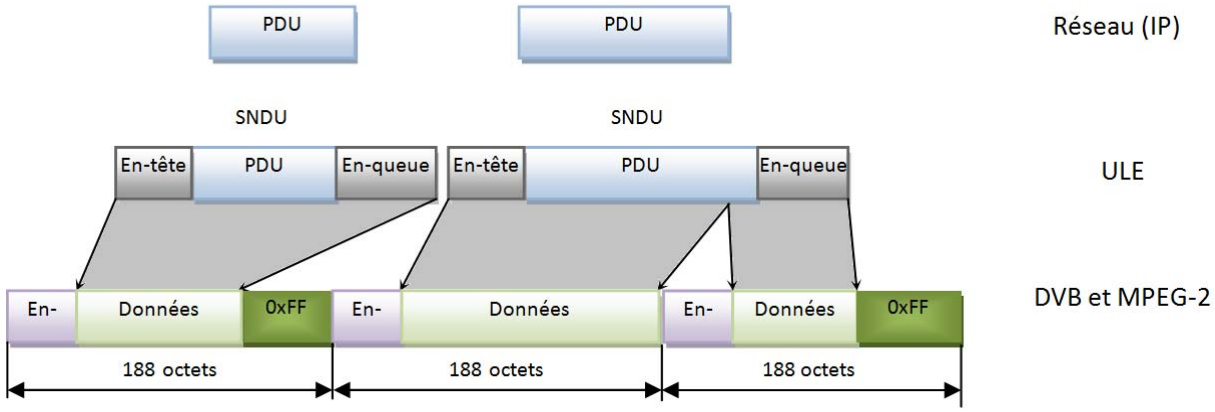


Figure 2.14: Encapsulation ULE avec padding

Le *packing* utilise les trames de manière plus efficace: s'il reste plus de deux octets après la fin d'un SNDU un nouveau SNDU commence dans la trame. Pour que cela soit possible, le drapeau PUSI de l'en-tête MPEG-2 est mis à 1 pour indiquer que la trame contient le début d'un SNDU et l'en-tête de 4 octets est suivi par l'adresse du début du SNDU, le PP. La procédure de *packing* est montrée dans la figure 2.15.

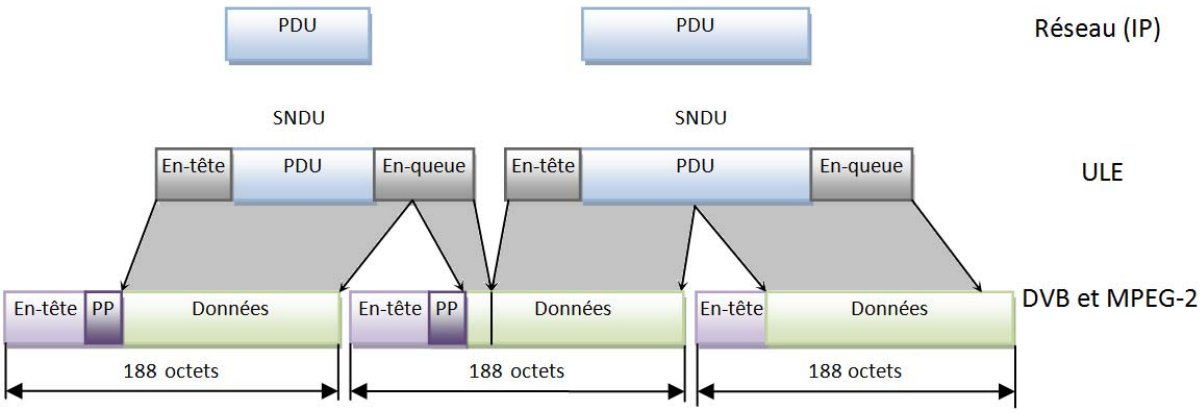


Figure 2.15: Encapsulation ULE avec packing

Le choix entre le *padding* et le *packing* est fait pour chaque système de communication en moment de son implémentation. Le *packing* permet des débits de données plus grands mais nécessite aussi des ressources de calcul plus importantes que dans le cas de la procédure *padding*.

2.4. Critères de sécurité (Cruickshank H., 2009, Iyengar S., 2007)

Les communications IP par DVB satellitaire sont des communications sans fil et, donc, sont facilement attaquables. En plus, aucun des standards décrits jusqu'ici n'incluent des services de sécurité.

Nous présentons dans ce paragraphe, les types d'attaques auxquelles les communications IP satellitaires sont exposées et les services de sécurité nécessaires pour les contrer.

Une analyse générale de la sécurité des communications IP avec encapsulation ULE a été faite par Cruickshank H. (2009). Le cas particulier des communications IP avec encapsulation ULE portées par satellite a été analysé par Iyengar S. (2007).

Pour ces types de communication, chaque nouveau système qui propose une nouvelle solution de sécurité doit tenir compte des critères de sécurité décrits entre autres, dans les deux documents cités ci-dessus.

2.4.1. Scénarios de menace

Dans le contexte des communications IP par satellite avec encapsulation ULE, les attaques actives possibles sont:

- Un encapsulateur IP, prétend être un autre. Ceci implique l'accès au fournisseur.
- La modification des messages d'une manière malveillante.
- La reproduction des messages. L'accès aux messages antérieurs est nécessaire.
- Attaques de type DoS (Denial of Service), c'est-à-dire, quant une entité est empêchée d'exécuter ses fonctions ou agit d'une manière qui empêche d'autres entités de fonctionner.

Les menaces actives présentées ci-dessus, sont la préoccupation principale des internautes, car dans les systèmes de communication IP classiques, leur réalisation n'est pas très compliquée. Cependant, dans le cas de systèmes de communication que nous analysons, l'implémentation des menaces en question est plus difficile. En effet, les trames MPEG-2 TS sont portées aux usagers par le système DVB-S, ce qui implique l'utilisation des codes FEC et d'une technique d'entrelacement de bits de plusieurs trames TS consécutives. En plus, l'antenne satellitaire du récepteur est très bien pointée vers le satellite relai et est accordé à la fréquence du fournisseur. C'est difficile pour un attaquant d'altérer les paquets de données originaux d'un certain ES (Elementary Stream) ou d'insérer ses propres paquets.

La menace la plus simple et la plus répandue, est la menace passive (écoute). C'est-à-dire un attaquant surveille les transmissions, afin d'obtenir soit l'information transmise elle-même, soit des informations relatives au trafic. Ce type de menace est la menace principale, car les équipements de réception ne sont pas très chers.

2.4.2. Requis de sécurité pour les communications IP par DVB satellitaire

Les services de sécurité nécessaires pour contrer les attaques passives et actives sont:

- Pour les attaques passives:
  - *Confidentialité des données:* service majeur pour protéger le contenu informatif.
  - *Protection des adresses NPA:* service permettant d'empêcher un attaquant de surveiller le volume du trafic d'un certain NPA (Network Point of Attachment).
- Pour les attaques actives:
  - *Intégrité des données et authentification de la source ULE.*
  - *Protection contre la reproduction des messages*.
  - *Authentification de la source et du récepteur:* réalisée pendant l'échange initial de clés, avant d'établir une liaison sécurisée entre le fournisseur et l'usager.

2.4.3. Considérations

L'analyse des attaques possibles contre les communications IP par DVB satellitaire que nous avons présentée, montre la nécessité de la sécurité au niveau ULE. Ce type de sécurité, qui est réalisée au niveau liaison des données (couche 2), ne doit pas remplacer les autres mécanismes de sécurité, tels que: l'IPSec (qui sera détaillé dans l'annexe A.2.) ou le TLS (Transport Layer Security) qui sont

réalisés aux niveaux supérieures, et tous les autres mécanismes qui sont implémentés hors niveau liaison des données. Il doit les renforcer.

2.5. Solutions existantes

Nous présentons dans ce paragraphe, les solutions existantes pour la sécurité des communications IP via DVB satellitaire et nous mettons en évidence leurs points faibles.

2.5.1. Extension de sécurité pour l'ULE (Cruickshank, 2008)

Le format de l'en-tête ULE est réduit au minimum pour avoir un minimum de données ajoutées et un minimum de puissance de calcul requise pour le traitement des SNDU. Si des services supplémentaires sont nécessaires, tel que la sécurité, le format ULE permet une ou plusieurs extensions de l'en-tête. Une solution de l'extension de l'en-tête, en vue de la sécurité, a été proposée par Cruickshank (2008). Cette extension tient compte de l'analyse de sécurité des communications IP satellitaire que nous avons présenté dans le paragraphe 2.4 et du format d'extension de l'en-tête, que nous avons présenté dans le paragraphe 2.3.3.2.

La figure 2.16, montre le format de cette extension d'en-tête:

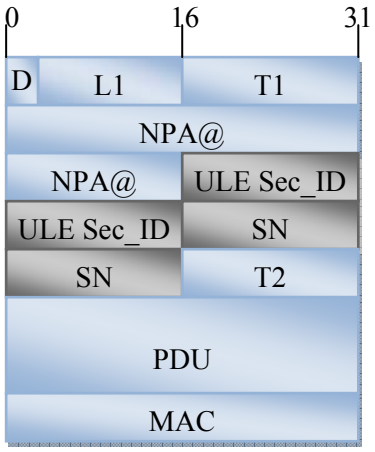


Figure 2.16: Extension de l'en-tête ULE en vue de la sécurité

Cette extension de l'en-tête, inspirée de l'IPSec, contient deux champs:

- *ULE Sec_ID (ULE Security Identifier)*: est un champ qui indique l'association de sécurité, de manière similaire que le SPI (Security Parameter Index) d'IPSec.
- *SN (Sequence Number*): est un compteur incrémenté pour chaque nouveau SNDU, permettant d'empêcher les attaques de type reproduction des messages *(replay)*.

Les services de sécurité que cette extension d'en-tête peut fournir sont:

- *Confidentialité de données*: service réalisé par le chiffrement des PDU.

- *Authentification de l'origine des données*: service réalisé par un code d'authentification des messages, le MAC (Message Authentication Code).
- *Iintégrité des données*: service réalisé aussi par le MAC, qui permet à l'utilisateur de vérifier que le message n'a pas été modifié tout au long de sa transmission.
- *Protection contre l'envoi multiple*: chaque SNDU a dans son extension d'en-tête, un nombre de séquences SN différent. Ceci permet au récepteur de ne traiter que les SNDU qui ont un nombre de séquences valide, ce qui empêche l'attaquant de réutiliser un ancien SNDU.

La solution de Cruickshank a deux inconvénients: le premier vient du fait que la solution proposée s'inspire trop d'IPSec, qui a été conçu pour les réseaux IP terrestres. Le deuxième inconvénient, tient du fait que la solution en question est incomplète, car elle ne traite pas un problème très important pour tout système de sécurité, le problème de la gestion des clés secrètes. Cruickshank recommande pour la gestion des clés, les techniques utilisées par IPSec. Cependant, il est clair que ces techniques de gestion des clés ne sont pas adaptées pour les systèmes de communication IP par satellite.

2.5.2. Sécurité niveau réseau

Une autre solution permettant d'offrir la sécurité aux communications IP par DVB satellitaire, est l'établissement d'un réseau virtuel privé entre le fournisseur des services IP (ISP) et le client. La solution la plus répandue est l'utilisation d'IPSec en mode tunnel.

Ceci implique des inconvénients tels qu'un taux des données ajoutées fort (jusqu'au 20% pour IPSec) et un retard important pour l'établissement des clés. Le retard important est dû au fait que l'établissement des clés est réalisé après que le fournisseur et le client aient changés au moins deux messages, ceci prend entre 1 à 1,5 secondes dans le cas des communications satellitaires.

2.6. Solution proposée

Dans ce paragraphe, nous décrivons en détails, la solution que nous proposons pour assurer la sécurité des communications IP par DVB-S. Notre solution de sécurité travaille au niveau ULE et intègre une large variété de composantes. En effet, nous proposons une extension de l'en-tête ULE, les algorithmes de chiffrement à utiliser, la procédure de gestion des clés, le transfert des données relatives à la sécurité et la structure de l'encapsulateur IP.

Nous montrons que toutes ces composantes sont compatibles et ensemble elles forment une solution de sécurité robuste, puis nous donnons tous les détails nécessaires en vue d'une implémentation pratique de la solution proposée.

2.6.1. Présentation générale

Nous proposons une solution de sécurité, basée sur les séquences chaotiques, qui tient compte des besoins de sécurité décrits dans le paragraphe 2.4, et des caractéristiques des communications IP par satellite en vue d'une implémentation simple et efficace. Le système de communication en question, est caractérisé par une asymétrie importante en largeur de bande entre la voie ascendante et la voie descendante et d'un temps de retard important. Pour cela, notre solution de sécurité utilise la voie descendante de manière plus intense que la voie ascendante et n'utilise pas d'échanges fréquents des messages entre l'utilisateur et le fournisseur. En fait, la plupart des messages de gestion des clés sont envoyés par le fournisseur et reçus par l'utilisateur. L'utilisateur envoie un message au fournisseur à propos des clés, seulement en cas de pertes de la synchronisation.

Nous rappelons que notre solution œuvre au niveau de la couche liaison des données: elle protège les trames SNDU ULE. A ce propos, nous proposons d'utiliser un champ PN (Packet Number) qui sera utilisé pour générer une clé de chiffrement différente pour chaque PDU et pour protéger contre les attaques actives. Aussi, nous proposons de chiffrer le code MAC pour une sécurité accrue. La génération des clés et le processus de chiffrement sont réalisés avec des algorithmes basés sur des fonctions chaotiques. Ceci permet d'avoir un générateur des clés secrètes et un algorithme de chiffrement qui possèdent de très bonnes propriétés cryptographiques.

L'authentification du terminal est réalisée avec un secret partagé, le système de gestion des clés multicouche. Ce système s'appuie sur une clé secrète partagée (avec un autre moyen de communication que la voie satellitaire, par exemple une carte à puce), entre le fournisseur des services Internet et le client.

2.6.2. Extension de la sécurité

La structure sécurisée du SNDU est présentée dans la figure 2.17. IANA doit assigner une valeur pour le champ T1, pour qu'il puisse identifier l'extension de l'en-tête.

Notre solution propose l'utilisation d'une extension de l'en-tête sur 32 bits avec le seul champ PN. Ce champ, inspiré du champ SN d'IPSec, offre une protection contre les attaques de type reproduction des messages et est utilisé comme nonce cryptographique pour la dérivation d'une nouvelle clé pour chaque paquet. Le chiffrement des PDU apporte la confidentialité des données. Pour ajouter aussi l'authentification et l'intégrité des données, nous proposons de remplacer le CRC du SNDU par un code MAC. Pour renforcer la sécurité du code MAC, nous proposons de le chiffrer (idée inspirée de la sécurité sur les WIFI, Géron. A., 2006). De cette manière, le code MAC protégera le PDU, l'en-tête ULE et permet de contrer les attaques actives.

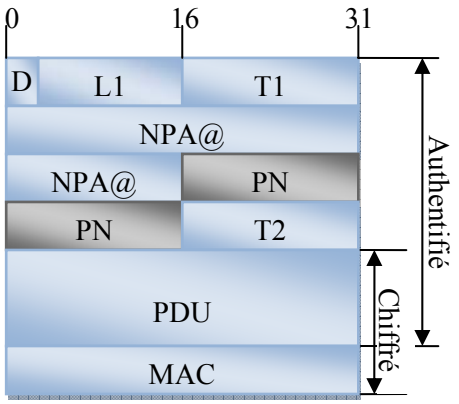


Figure 2.17: SNDU sécurisé selon la solution de sécurité proposée.

2.6.3. Algorithme de chiffrement proposé

La solution de sécurité que nous proposons, pour l'encapsulation ULE, permet l'utilisation de plusieurs algorithmes de chiffrement pour la protection des données. En fait, l'algorithme de chiffrement, ainsi que la clé secrète peuvent changer très fréquemment. Nous allons montrer le mécanisme de changement des algorithmes dans le paragraphe 2.6.5.1, pour les communications unicast et dans le paragraphe 2.6.5.2, pour les communications multicast.

Par ailleurs, nous recommandons qu'un de ces algorithmes de chiffrement ci-dessous soit utilisé : ECKBA modifié (Enhanced 1-D Chaotic Key-Based Algorithm, Socek, 2005, Caragata, 2006) ; DFRCBCSIM (Design of a Fast and Robust Chaos-Based Crypto-System for Image Encryption, Noura 2011). Aussi des algorithmes publics qui sont considérés robustes de point de vue cryptographique, tel que l'AES (Advanced Encryption Standard, NIST, 2001b).

En ce qui suit, nous présentons l'algorithme ECKBA modifié.

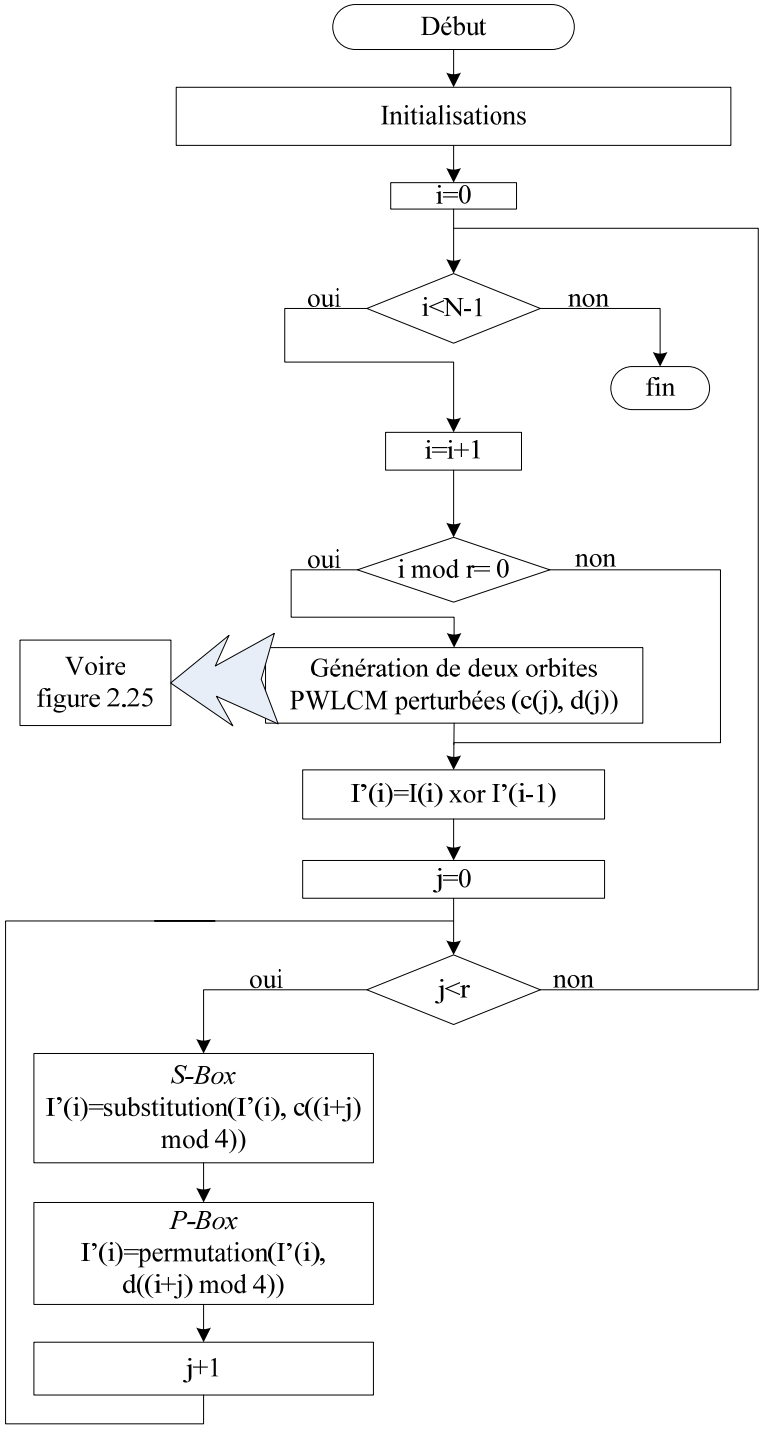


Figure 2.18: Algorithme de chiffrement ECKBA modifié.

L'algorithme ECKBA modifié a été proposé pour le chiffrement des images, mais il peut être utilisé aussi pour chiffrer les flux des SNDU. C'est un algorithme de chiffrement par bloc : il divise le flux à chiffrer en blocs de longueur fixe (8 bits) et fait le chiffrement de chaque bloc en mode CBC. L'algorithme utilise une clé secrète à 128 bits et la fonction chaotique PWLCM. Il utilise une fonction de permutation *P-Box* pour la diffusion et une fonction de substitution *S-Box* pour ajouter de la confusion au système. Les deux processus forment un réseau *SP-Box* et sont répétés *r* fois pour chaque bloc.

Les principales opérations effectuées par l'algorithme sont :

- Génération, grâce à deux cartes chaotiques PWLCM, des valeurs chaotiques *c()* et *d()* pour les processus de confusion (*S-box*) et de diffusion (*P-box*). L'initialisation de deux cartes chaotiques utilise une clé secrète de 128 bits (64 bits par carte chaotique).
- Perturbation des orbites *c()* et *d()* afin de diminuer l'effet de la dégradation dynamique causé par la quantification sur N=32 bits.
- Utilisation du mode de chiffrement avec chaînage de blocs CBC (Cipher Block Chaining), où le chiffrement du bloc informatif courant utilise le bloc chiffré précédemment.
- Itérations multiples (rounds *r*) des processus de substitution (confusion) et de permutation (diffusion).

La clé de 128 bits est décomposée comme suit :

32 bits Condition initiale pour la première carte chaotique PWLCM	**32 bits** Condition initiale pour la deuxième carte chaotique PWLCM	**32 bits** Paramètre contrôle de la première carte chaotique PWLCM	**32 bits** Paramètre contrôle de la deuxième carte chaotique PWLCM

Figure 2.19: format de la clé secrète de l'algorithme ECKBA modifié

Un des principaux problèmes des implémentations numériques des cartes chaotiques, est la dégradation dynamique due à la représentation en précision finie. Pour contrer ce problème, nous proposons une technique de perturbation de l'orbite chaotique, basée sur l'utilisation d'un LFSR, El Assad (2008). Le principe de la méthode de perturbation est montré dans la figure 2.20 :

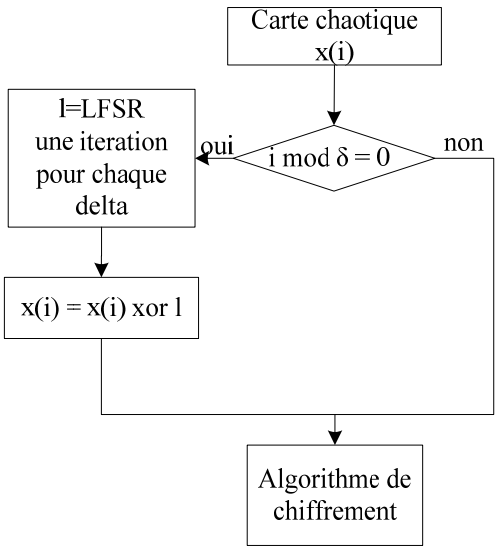


Figure 2.20: Technique de perturbation utilisant un LFSR

2.6.4. Système proposé de gestion des clés multicouche

Les communications satellitaires peuvent être de trois types:

- *Broadcast*: le message est adressé à tous les récepteurs situés dans la zone de couverture du satellite. C'est le cas de la diffusion de programmes de télévision par satellite. Bien sûr, ces communications n'ont pas besoin d'être sécurisées.
- *Unicast*: le message est adressé à un seul utilisateur. C'est le cas de la navigation IP par satellite.
- *Multicast*: le message est adressé à un groupe des récepteurs. C'est le cas de la télévision payante ou de certains services IP par satellite, comme les vidéo conférences.

La solution de sécurité que nous proposons, s'adresse aux communications unicast et multicast.

Dans l'annexe A.3, nous présentons différents protocoles de gestion des clés. Pour notre solution de sécurité, nous proposons l'utilisation d'un système de gestion des clés multicouche. Dans ce cas, tous les PDU sont chiffrés utilisant une clé de session SK (Session Key). Cette clé est utilisée avec le nonce cryptographique PN pour dériver une clé éphémère EK, qui sera utilisé comme clé secrète de l'algorithme de chiffrement. Chaque clé de session est utilisée pour chiffrer une quantité limitée d'information, définie par le paramètre SKTh (Session Key Threshold). Quand la clé de session doit être changée, une nouvelle clé de session est générée par le fournisseur et est envoyée chiffrée avec la clé de niveau supérieur IK1 (Intermediary key level 1) à l'utilisateur concerné. De même, la clé IK1 est utilisée un nombre limité de fois, défini par le paramètre IK1Th. Quand elle doit être changée, une nouvelle valeur est générée par le fournisseur et est envoyée chiffrée avec la clé de niveau supérieur IK2. Et ainsi de suite, c.à.d, en général, la clé IK*i* est utilisée un nombre limité de fois, défini par le paramètre IK*i*Th. Quand la clé en question doit être changée, le fournisseur génère une nouvelle valeur pour IK*i* et l'envoie chiffré, à l'utilisateur concerné, avec la clé IK*(i+1)*. Dans ce système, le niveau le plus haut concerne la clé MK (Master Key), qui est la seule clé qui n'est pas changée par la voie satellitaire. Elle est convenue entre le fournisseur et les usagers par d'autres moyens de communication, (abonnement et carte à puce).

Ce système est représenté dans la figure ci-dessous:

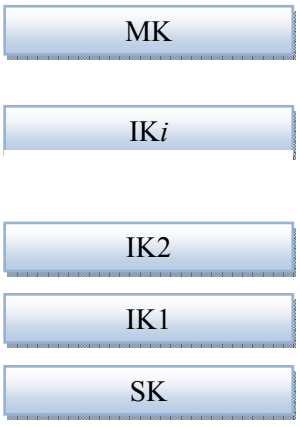


Figure 2.21: Système de gestion des clés multicouche

2.6.4.1. Cas des communications Unicast

Le SNDU est sécurisé en utilisant la clé de session SK. La clé SK doit être changée aussi fréquemment que possible, alors que la clé MK devrait être changée seulement en cas de compromission. Nous proposons l'utilisation d'une longueur de 128 bits pour SK pour suivre les recommandations d'Ecrypt II (2010). Bien sûr, d'autres valeurs peuvent être utilisées. Si la longueur de la clé augmente, la sécurité des données augmente avec elle, mais, le coût du traitement des données augmente aussi.

La clé MK, doit avoir une taille importante, car la sécurité du système entier en dépend. Pour cela nous proposons l'utilisation d'une taille d'au moins de 1024 bits afin de rendre pratiquement impossible l'attaque exhaustive.

En général, dans la hiérarchie multicouche, il y a une série de clés entre la couche de niveau le plus haut (clé MK) et la couche de niveau le plus bas (clé SK). Leur nombre et leur taille peuvent être choisis par le fournisseur, selon la structure du système et la politique de sécurité adoptée. Nous proposons et analysons, un système de gestion multicouche, des clés, à 3 niveaux intermédiaires, comme montré dans la figure 2.22. Nous utilisons une clé intermédiaire niveau 1 (IK1) ayant une taille de 256 bits, une clé intermédiaire du niveau 2 (IK2) ayant une taille de 512 bits et une clé intermédiaire de niveau 3 (IK3) ayant une taille de 1024 bits. Le choix adopté de la taille des 3 clés intermédiaires, assure un très haut niveau de sécurité et nous permet d'étudier un cas très défavorable de point de vue du taux des données ajoutées. Nous allons montrer, que malgré l'utilisation de 3 clés intermédiaires, le taux des données ajoutées par notre solution de sécurité est très faible comparé à d'autres solutions.

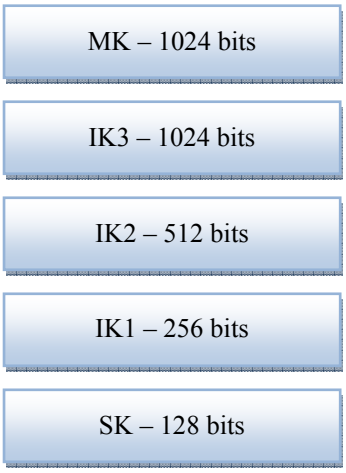


Figure 2.22: Structure du système de gestion des clés proposé

2.6.4.2. Cas des communications Multicast

Dans le cas des communications multicast par satellite, il y a un contrôleur de group GC (Group Controller) qui gère les communications, et il y a un nombre limité des membres GM (Group

Members) qui ont accès à l'information. Bien sûr, pour notre cas, le GC est le fournisseur des services IP et les GM sont les utilisateurs.

Différentes solutions pour la gestion des clés dans les systèmes de communication multicast ont été proposées, mais toutes utilisent la même structure. Chaque GM partage une clé secrète individuelle PWK (Pair Wise Key) avec le GC. Par ailleurs, tous les GM partagent une clé commune TEK (Traffic Encryption Key) avec le GC, clé qui sert pour le chiffrement des données. Aussi, dans la plupart des communications multicast, il y a un nombre des clés intermédiaires KEK (Key Encryption Keys) qui sont utilisées dans le processus de gestion des clés.

Howarth M. P. (2004) a montré que le système de gestion des clés le plus adapté pour le multicast satellitaire est le système des arbres des clés LKH (Logical Key Hierarchy). La structure générale de l'LKH est montrée dans la figure 2.23.

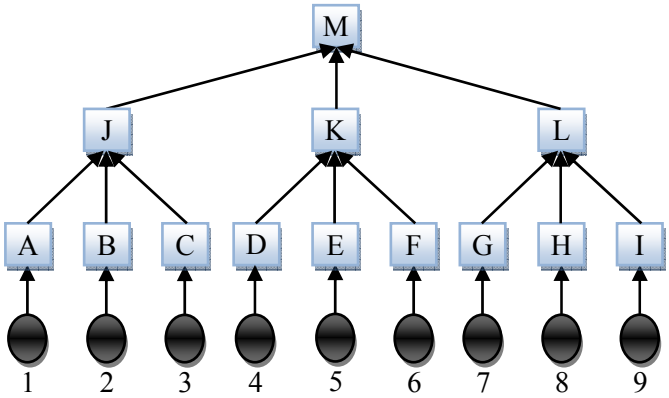


Figure 2.23: Structure de l'arbre des clés LKH

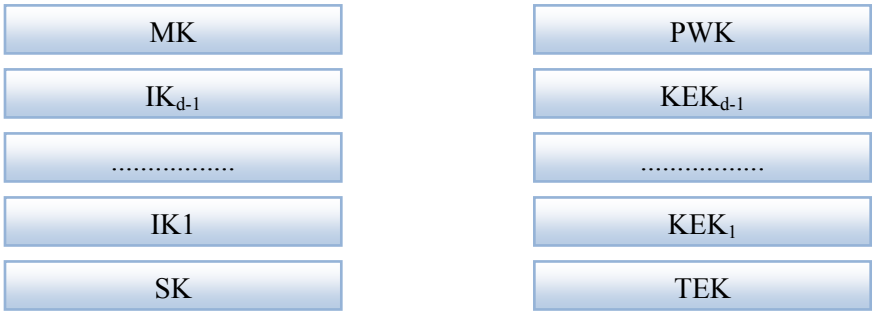
Le nombre de couches des clés est variable en fonction du nombre d'utilisateurs et de la politique de sécurité du système. Nous présentons un exemple avec trois couches des clés.

Les membres (utilisateurs) GM sont représentés par des cercles de couleur noire, numérotés de *1* à *9*. Chaque membre GM partage une clé PWK avec le GC. Ces clés sont indiquées par les lettres *A* à *I* dans la figure 2.23. Les clés *J*, *K* et *L* sont des clés intermédiaires, les KEK. La clé *M* est la clé TEK, utilisée pour le chiffrement des données. Chaque GM connait les clés qui forment une liaison entre sa PWK et la TEK. Par exemple un membre GM 9 connait la PWK *I*, la KEK *L* et la TEK *M*.

Si un certain membre, par exemple le GM 9, doit quitter le groupe, le GC doit changer toutes les clés le concernant, à l'exception de sa propre clé PWK. Pour cela et en premier, le GC procède à la génération d'une nouvelle valeur pour la clé *L*. Puis, il envoie la valeur chiffrée avec la clé *G* au membre GM *7* et la valeur chiffrée avec la clé *H* au membre GM *8*. Ensuite, le GC génère une nouvelle valeur pour la clé *M*. Puis, il envoie cette valeur chiffrée: avec la clé *J,* aux membres GM *1*, GM *2* et GM3 avec la clé *K,* aux membres GM *4*, GM *5* et GM *6*; et enfin, avec la clé *L,* aux membres GM *7* et GM *8*.

Si un nouveau membre doit rejoindre le group de multicast, par exemple un membre rejoint le GM *9*, alors, toutes les clés auxquelles il aura accès doivent être mises à jour. Pour ce faire, le même mécanisme que nous venons de présenter sera utilisé.

Si le système de communications IP par DVB-S est utilisé pour des communications multicast, alors, le fournisseur est le GC et les clients sont les GM. Le fournisseur génère tout l'arbre des clés et chaque client aura connaissance des clés le concernant, à savoir les clés qui forment une liaison entre sa PWK et la TEK. Du point de vue des utilisateurs, cette structure est identique avec la structure multicouche utilisée pour les communications unicast, comme c'est montré par la figure 2.24.



(a) *b)* .

Figure 2.24. Similarité des structures des clés unicast et multicast:

(a) clés du système unicast, (b) clés du système multicast.

2.6.4.3. Générateur des clés

Nous proposons pour la génération des clés secrètes SK, IK1, IK2 et IK3, un générateur chaotique proposé par El Assad (2008) et Noura (thèse en cours). Le générateur proposé est constitué de deux filtres récursifs IIR (Infinite Impulse Response) en parallèle, contenant chacun une fonction non linéaire comme indiqué dans la figure 2.25.

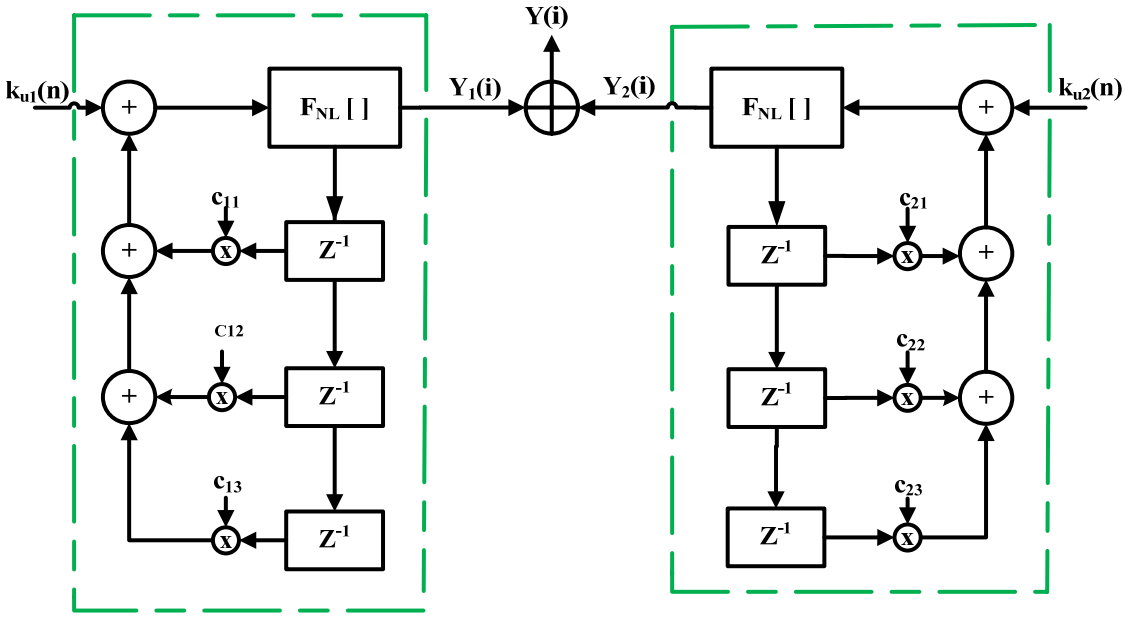


Figure 2.25: Générateur des clés proposé

Le bloc F_{NL} est une fonction non linaire. Plusieurs fonctions non linéaires ont été testées: *x·ln(x), x·exp(cos(x)), Skew Tent map, PWLCM map (Piecewise Linear Chaotic Map), etc.* Les coefficients c_{11}, c_{12}, c_{13}, c_{21}, c_{22}, et c_{23} peuvent prendre des valeurs comprises entre *1* et 2^N*-1*, où *N* est le nombre des bits de quantification. Le générateur est défini par les relations suivantes:

$$Y_1(n) = FNL(k_{u1}(n) + \sum_{i=1}^{3}[c_{1i} \times Y_1(n-i)] \tag{2.1}$$

$$Y_2(n) = FNL(k_{u2}(n) + \sum_{i=1}^{3}[c_{2i} \times Y_2(n-1)] \tag{2.2}$$

$$Y(n) = Y_1(n) \oplus Y_2(n) \tag{2.3}$$

Les entrés $k_{u1}(n)$ et $k_{u2}(n)$ sont des paramètres supplémentaires du générateur. Le système a été implémenté avec une précision de 32 bits, alors 4 itérations sont nécessaires pour générer une clé a 128 bits.

Les travaux menés par El Assad et Noura ont montré que les meilleurs propriétés cryptographiques du générateur en question sont obtenues avec les fonctions non-linéaires *PWLCM* et *Skew Tent*. La fonction *Skew Tent map* est composé de deux segments linaires donnés par:

$$y(n) = F[y(n-1)] = \begin{cases} \dfrac{y(n-1)}{p} \; if \; 0 \le y(n-1) < p \\ \dfrac{1-y(n-1)}{1-p} \; if \; p \le y(n-1) < 0.5 \end{cases} \tag{2.4}$$

Où p est le paramètre de contrôle tel que: $\mathbf{0} < p < 1$.

2.6.4.4. Dérivation des clés éphémères

Le chiffrement et l'authentification sont réalisés avec des clés EK (Ephemeral Key) propres à chaque PDU. Ces clés sont obtenues à partir d'une fonction de hachage dont les entrées sont: la clé SK pour les communications unicast ou la clé TEK pour les communications multicast, l'adresse NPA si elle est utilisée ou bien le PID, si l'adresse NPA n'est pas utilisée et le nonce PN. L'utilisation du PN garantie qu'une nouvelle clé EK sera générée pour chaque nouveau PDU.

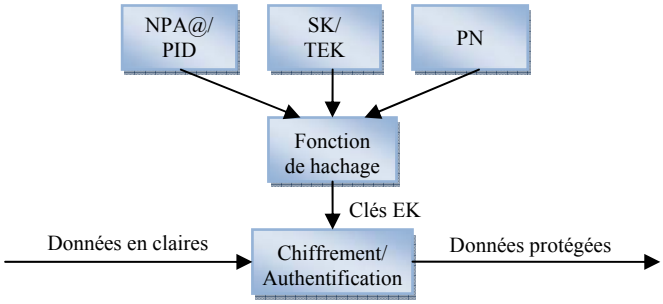


Figure 2.26: Calcul des clés éphémères.

Nous pouvons remarquer la compatibilité entre les communications unicast et multicast. En effet, pour passer du système de communications unicast au système de communication multicast, il suffit, afin de générer les clés éphémères, de remplacer la clé SK par la clé TEK.

Par ailleurs, notre solution de sécurité inclut un mécanisme (présenté dans le paragraphe suivant) permettant le changement dynamique: de l'algorithme de chiffrement, de l'algorithme d'authentification et de la fonction de hachage utilisée. Ceci a pour conséquence d'augmenter de manière significative la sécurité des communications.

2.6.5. Format proposé des données concernant les paramètres de sécurité

Une partie importante de la solution de sécurité proposée, est la transmission des données concernant les paramètres de sécurité. Nous proposons, en ce qui suit, deux nouveaux types de PDU, qui sont utilisés pour la transmission des données, concernant les paramètres de sécurité pour les communications unicast ou multicast. Aussi, nous discutons le mécanisme qui permet le rétablissement de la synchronisation des clés si la synchronisation en question est perdue.

2.6.5.1. Cas des communications Unicast

Pour les communications unicast, un nouveau type de PDU, le SPDU (Security PDU), est nécessaire pour permettre la transmission de l'information de gestion des clés, à savoir: les nouvelles clés multicouche, les algorithmes de chiffrement à utiliser, les algorithmes d'authentification et la procédure de dérivation des clés EK.

La structure du SPDU, donnée par figure 2.27, est similaire avec la structure de n'importe quel PDU: un en-tête suivi par le corps du message. L'en-tête contient l'information relative à l'Association de Sécurité (AS) créé et le corps du message contient les clés sécrètes chiffrées.

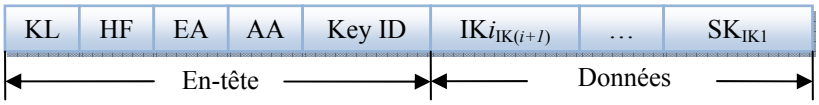


Figure 2.27: Structure de la trame SPDU.

L'en-tête contient les champs suivants:

- *KL (Key Level)*: un champ sur 2 bits qui indique les clés multicouche transportées par le SPDU: soit SK seulement, soit IK1 et SK, soit IK2, IK1 et SK, soit IK3, IK2, IK1, et SK. Bien sur, ces clés sont chiffrées.
- *HF (Hash Function):* un champ sur 2 bits qui indique la fonction de hachage utilisée pour la dérivation des clés EK.
- *EA (Encryption Algorithm):* un champ sur 2 bits qui indique l'algorithme de chiffrement qui sera utilisé. Cet algorithme sera utilisé pour le chiffrement de toutes les données envoyées sur la voie satellitaire: les données utilisateur, les données de sécurité, autres données qui peuvent être portées par les trames SNDU ULE.
- *AA (Authentication Algorithm):* un champ sur 2 bits qui indique l'algorithme d'authentification à utiliser.
- *Key ID:* un champ sur 8 bits qui identifie l'association de sécurité qui vient d'être créé. Cette valeur sera utilisée en cas de pertes de la synchronisation entre le fournisseur et le client. Dans ce cas, l'utilisateur envoie un message au fournisseur indiquant le dernier *Key ID* synchronisé.

Le corps du message contient les nouvelles clés. Dans la figure 2.27, $IKi_{IK(i+1)}$ est un des IK1, IK2 ou IK3 chiffré avec IK2, IK3 ou MK.

Le SPDU doit être encapsulé comme n'importe quel PDU. Pour que cela soit possible, IANA doit assigner une valeur du champ *T1* qui indique un SPDU.

2.6.5.2. Cas des communications Multicast

Pour les communications multicast, nous proposons l'utilisation d'un autre type de PDU, le MPDU (Multicast PDU). Cette nouvelle structure de trame est semblable à la structure de la trame SPDU de la figure 2.27, sauf la taille du champ KL et les clés du chiffrement transportées. En effet, la taille du champ KL dans le cas multicast est plus importante que dans le cas unicast. La taille exacte dépendra de l'application envisagée (exemples : en télévision payante la taille du champ KL est nettement plus grande que dans le cas de la visioconférence). Aussi, des clés du chiffrement KEK*i* seront transportées à la place des clés IK*i*. La figure 2.28, montre la structure du MPDU.

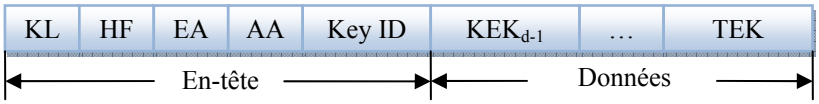


Figure 2.28: Structure de la trame MPDU.

2.6.5.3. Message d'alarme

Dans des circonstances inhabituelles, telles que: un niveau élevé de bruit, une panne matérielle, une panne de courant, des attaques actives ou autres raisons, le client peut perdre la synchronisation des clés avec le fournisseur. Dans ce cas, il sera incapable de déchiffrer les messages reçus. Pour rétablir la synchronisation, le client envoie un message spécial, que nous appelons *message d'alarme* au fournisseur indiquant le *Key ID* du dernier AS synchronisé. Ceci permettra au fournisseur d'établir une nouvelle AS, et ainsi la synchronisation.

Nous ne discutons pas ici la structure du message d'alarme, car il est envoyé par le client sur la voie ascendante. Nous rappelons que nous étudions dans ce chapitre seulement la voie descendante et que nous ne mettons aucune restriction sur la voie ascendante.

2.6.6. Modification du système

La solution de sécurité proposée, doit être implémentée dans l'encapsulateur IP pour le fournisseur et dans l'IPPRU pour le client. Des structures possibles de ces équipements sont présentées dans les figures 2.29 et 2.30. Elles intègrent, par rapport au système standard, des fonctions: de génération et de gestion des clés, de chiffrement et d'authentification. La structure que nous présentons n'est pas obligatoire.

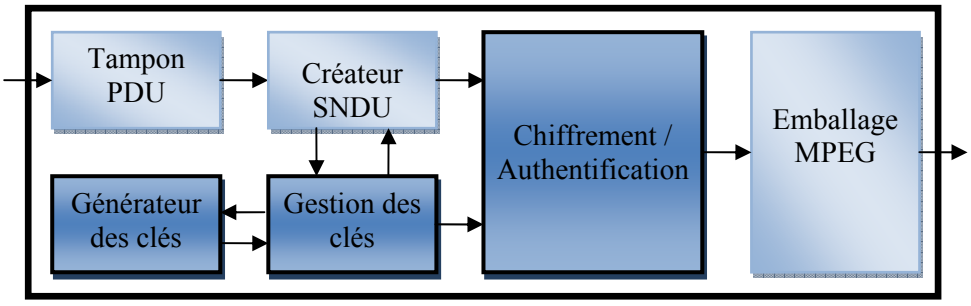


Figure 2.29: Structure de l'encapsulateur IP sécurisé

Dans la structure de l'encapsulateur IP sécurisé, le bloc *Gestion des clés* est le bloc le plus important. Son rôle est de:

- Commander le bloc *Générateur des clés,* qui crée les SPDU ;
- Communiquer avec le bloc *Créateur SNDU,* pour lui fournir le PN ;

- Fournir l'EK et les type d'algorithmes de chiffrement et d'authentification au bloc *Chiffrement/Authentification*.

Le bloc *Chiffrement/Authentification,* traite le code MAC pour l'en-tête et le PDU, et réalise le chiffrement du PDU et du code MAC. Il reçoit le type d'algorithmes et les clés de chiffrement et d'authentification du bloc *Gestion de clés*.

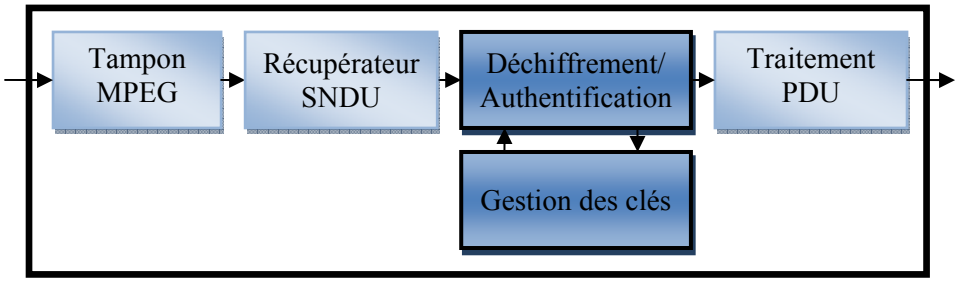


Figure 2.30: Structure du IPPRU sécurisé

Dans la structure de l'IPPRU *sécurisé* proposé, le bloc de *Déchiffrement/Authentification* envoie les SPDU et les PN au bloc *Gestion des clés* qui gère la structure des clés et traite l'EK.

2.7. Analyse des performances du système proposé

Dans ce paragraphe, nous analysons les performances du système proposé, selon le critère du taux des données ajoutées et selon la fréquence d'utilisation de la clé master MK.

La simulation, de la voie satellitaire pour les communications IP sécurisé et l'analyse des résultats obtenus seront présentés dans le paragraphe 2.8.2.

L'analyse du système de gestion des clés présenté dans la figure 2.22, est faite avec 5 jeux de valeurs (numérotés de I à V) pour les paramètres SKTh, IK1Th, IK2Th et IK3Th. Ces 5 jeux de valeurs ont été choisis de sorte à imposer une fourchette de la fréquence d'utilisation de la clé MK entre 6 heures et 2 ans, et ceci pour un débit des données allant de 256 Kbps à 90 Mbps. L'intervalle de débit choisi, couvre les différents débits possibles qu'un usager est amené à utiliser: Débit minimal de 256 kbps, débit moyen de l'ordre de 1 Mbps et débit max égale à 90 Mbps.

Le tableau 2.1, montre les 5 jeux de valeurs utilisées des paramètres.

Tableau 2.1 - Jeux de valeurs analysées des paramètres.

	I	**II**	**III**	**IV**	**V**
IK3Th	50	100	200	500	1000
IK2Th	50	100	200	500	1000
IK1Th	50	100	200	500	1000
SKTh	256 kb	512 kb	1 Mb	2 Mb	4 Mb

Il va de soit que d'autres jeux de valeurs des paramètres peuvent être utilisés. Le paramètre SKTh peut aussi s'exprimer en nombre de fois que la clé SK doit être utilisée, au lieu de la quantité d'information à sécuriser, comme indiquée dans le tableau 2.1. Les jeux des paramètres que nous avons choisis nous permettent de faire une analyse concrète de la solution de sécurité proposée. Cette analyse permet, d'un coté de prouver que les caractéristiques de la solution proposée sont très bonnes et, d'autre côté, de fournir toute l'information nécessaire à un fournisseur de services IP par satellite lui permettant de choisir les jeux de valeurs adéquates des paramètres pour une liaison donnée.

Le taux des données ajoutées est un paramètre important pour un système de communications par satellite, puisque, dans ce cas, la ressource spectrale est rare et chère, et elle doit être utilisée de manière très efficace. En effet, un nombre limité (quelques dizaines) des transpondeurs, qui ont une largeur de bande limitée approximativement à 40 MHz, sont disponibles. Donc, pour que le système de communications soit fiable du point de vue économique, la ressource spectrale doit être utilisée de manière optimale.

Un autre paramètre qui doit être pris en compte, est la fréquence d'utilisation PMK (Period of MK usage) de la clé MK. Il est important d'analyser ce paramètre car son utilisation de façon équilibrée est importante. La définition exacte du mot « équilibré » dépend du niveau de sécurité assuré par le fournisseur et du taux des données ajoutés désiré.

Nous analysons par la suite, les performances obtenues lors de l'utilisation de valeurs PMK comprises entre 6 heures et deux ans.

2.7.1. Taux des données ajoutées

Le taux des données ajoutées DO (*Data Overhead*), représente le taux des données supplémentaires envoyées par la voie satellitaire pour assurer le service de sécurité. Le taux DO est exprimé comme le rapport (en %) entre l'information supplémentaire ajoutée et l'information totale envoyée, et il est donné par la formule (*2.5*) suivante.

$$DO = \frac{AI}{TI} \cdot 100 \qquad (2.5)$$

où *AI* (*Added Information*) est la quantité d'information ajoutée par la solution de sécurité et *TI (Total Information)* est la quantité totale d'information envoyée.

Pour notre solution de sécurité, *AI* possède deux composantes EHI et KMI:

- *EHI (Extension Header Information):* la composante d'extension de l'en-tête, composée des octets de l'extension de l'en-tête pour tous les SNDU transportés par satellite.
- *KMI (Key Management Information) :* la composante de gestion des clés, composée des SNDU qui transportent des SPDU. Nous considérons que ces SNDU ont un en-tête standard puisque la composante de l'extension de l'en-tête est déjà prise en compte par l'*EHI.*

Ceci implique :

$$DO = \frac{AI}{TI} \cdot 100 = \frac{KMI + EHI}{TI} \cdot 100 = \frac{KMI}{TI} \cdot 100 + \frac{EHI}{TI} \cdot 100 \qquad (2.6)$$

Le premier terme, de l'expression précédente, noté DO_{KM}, et rappelé par la relation (2.7), varie en fonction de jeux de valeurs des paramètres de sécurité donnés par le tableau 2.1, et de la longueur moyenne des PDU.

$$DO_{KM} = \frac{KMI}{TI} \cdot 100 \qquad (2.7)$$

Les différents résultats présentés dans ce qui suit, ont été obtenus sous Matlab. La figure 2.31 montre, les résultats obtenus pour DO_{KM}, en fonction de la longueur moyenne des PDU et ceci pour chaque jeu de valeurs des paramètres.

Remarque: tous les termes ci-dessus sont mesurés en octets.

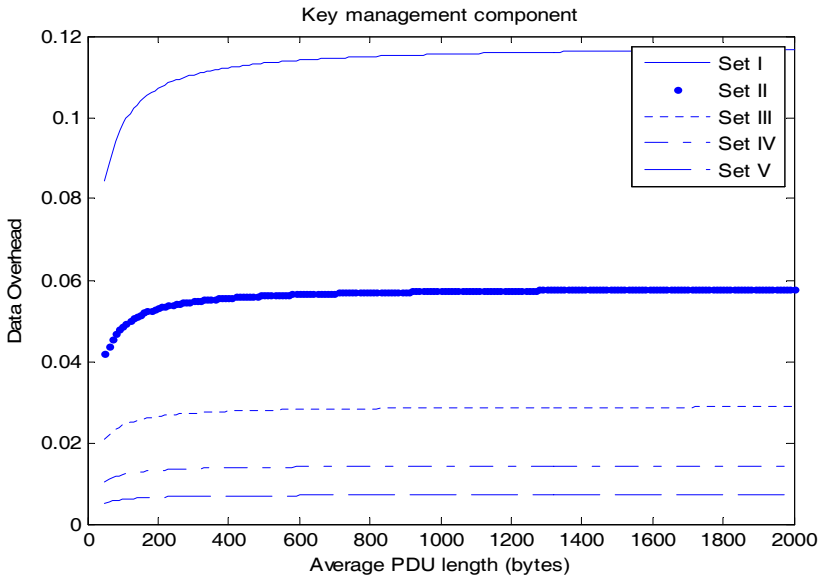


Figure 2.31: Résultats de simulation du DO_{KM}.

Nous pouvons remarquer d'après les résultats obtenus, que la valeur maximale du DO_{KM} est inferieure à 0.12% et que l'augmentation de la longueur moyenne des PDU implique une très légère augmentation du DO_{KM}. Dans le tableau 2.1, les valeurs des paramètres sont doublées d'un jeu à l'autre, et nous pouvons remarquer sur la figure 2.31 que la valeur de DO_{KM} est doublée aussi d'un jeu à l'autre.

Le deuxième terme de l'expression (2.6), noté DO_{EH}, est rappelé par la relation (2.8) suivante.

$$DO_{EH} = \frac{EHI}{TI} \cdot 100 \qquad (2.8)$$

Avec:

$$EHI = N \cdot EH \tag{2.9}$$

$$TI = N \times l_{SNDU} \tag{2.10}$$

Où N est le nombre des SNDU, *EH* est la longueur de l'extension de l'en-tête (4 octets) et l_{SNDU} est la longueur moyenne des SNDU.

Finalement, d'après les équations (2.9) et (2.10), la relation (2.8) devient :

$$DO_{EH} = \frac{EH}{l_{SNDU}} \cdot 100 \tag{2.11}$$

DO_{EH} dépend uniquement de la longueur moyenne des PDU, car la même extension de l'en-tête est utilisée pour les SNDU qui portent les PDU des données et pour les SNDU qui portent les SPDU.

La figure 2.32, montre l'évolution du DO_{EH} en fonction de la longueur moyenne des PDU.

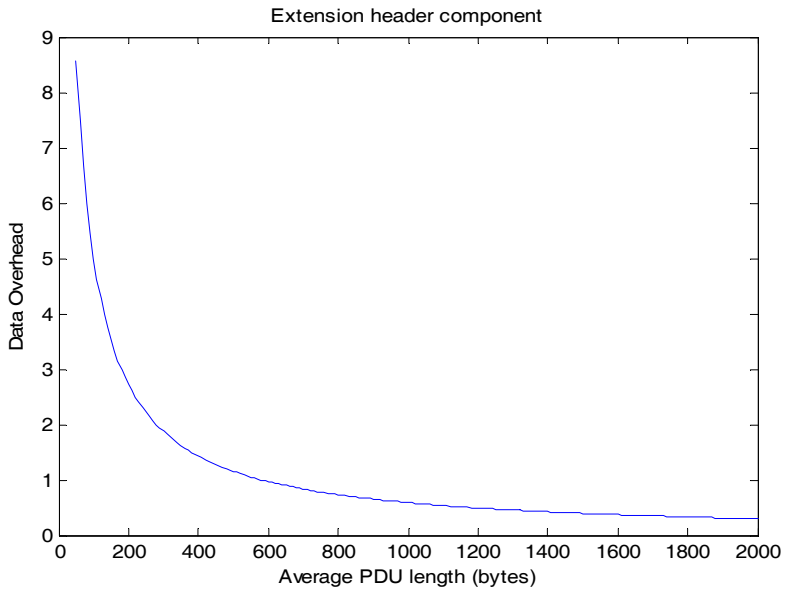
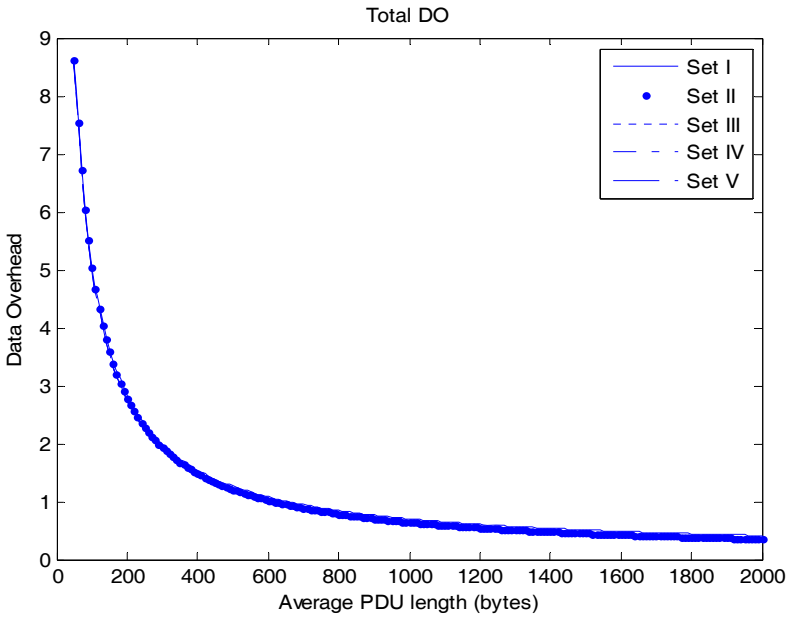


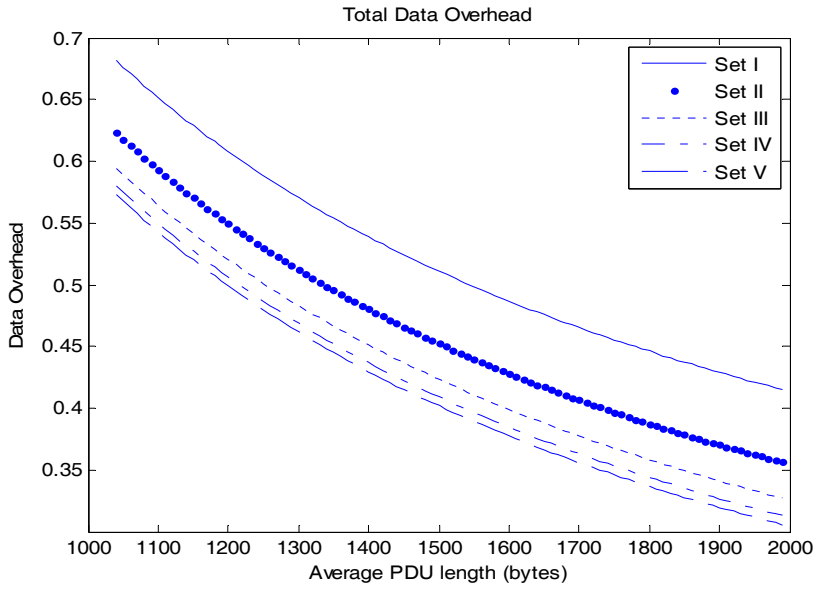
Figure 2.32: Evolution du terme DO_{EH}.

Nous remarquons, un taux des données ajoutées relativement grand, pour les paquets avec une longueur plus petite que 200 octets, et un taux des données ajoutées relativement faible, pour les paquets de longueur plus grande que 600 octets.

Enfin, les courbes de la figure 2.33 *a) et b)* donnent le taux total des données ajoutées DO.



(a)



(b)

Figure 2.33: Taux total des données ajoutées (a) pour des longueurs des PDU entre 50 et 2000 octets; (b) pour des longueurs des PDU entre 1000 et 2000 octets.

2.7.2. Période d'utilisation PMK, de la clé MK

L'étude de la période d'utilisation PMK, de la clé MK, est importante, car elle permet au fournisseur de choisir le jeu adéquat des paramètres de sécurité, qui assure la politique de sécurité adoptée avec un coût minimum.

Le PMK est calculé par la formule suivante:

$$PMK = \frac{TI}{Bitrate} \tag{2.12}$$

où *Bitrate* est le débit du canal satellitaire et *TI* est la quantité des données envoyée entre deux utilisations successives de la clé MK.

Le tableau 2.2, montre les différentes valeurs du PMK exprimées en jour, en fonction du débit utilisé et ceci pour chaque jeu de valeurs des paramètres du tableau 2.1. Seulement les valeurs plausibles, entre 0.25 jours (6 heures) et 730 jours (2 ans) sont mentionnées.

Tableau 2.2 – Période d'utilisation PMK de la clé MK.

	256 kbps	**1 Mbps**	**5 Mbps**	**20 Mbps**	**90 Mbps**
I	1,52	0,38			
II	23,68	5,92	1,18	0,29	
III	372,6	93,17	18,63	4,66	1,04
IV				144,1	32,04
V					510,9

2.8. Simulations et résultats

2.8.1. Robustesse du générateur des clés proposé

La protection des clés est une partie vitale pour tout système assurant la sécurité des données. Nous avons testé le générateur des clés proposé pour vérifier si les clés générées possèdent de bonnes propriétés cryptographiques ou non. Dans les figures 2.34, 2.35 et 2.36 nous présentons quelques résultats permettant de quantifier les performances du générateur proposé.

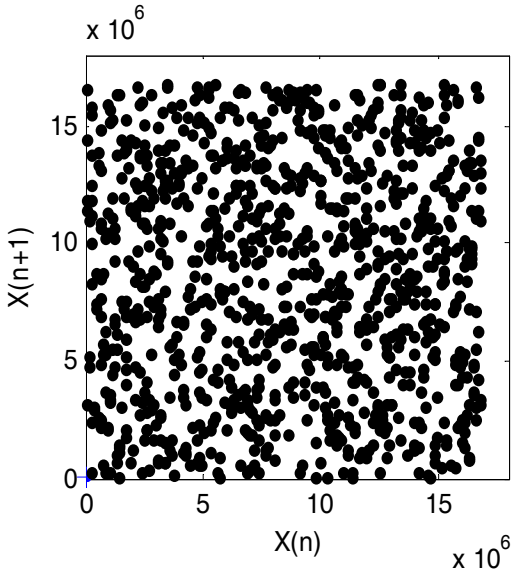


Figure 2.34: Résultat du mappage.

Le résultat du mappage, présenté dans la figure 2.34, montre clairement une distribution quasi aléatoire des séquences générées.

Dans la figure 2.35 nous montrons un exemple de résultats obtenus pour les fonctions d'auto et d'inter-corrélation entre séquences générées. Ces résultats sont pratiquement semblables à ceux d'un bruit.

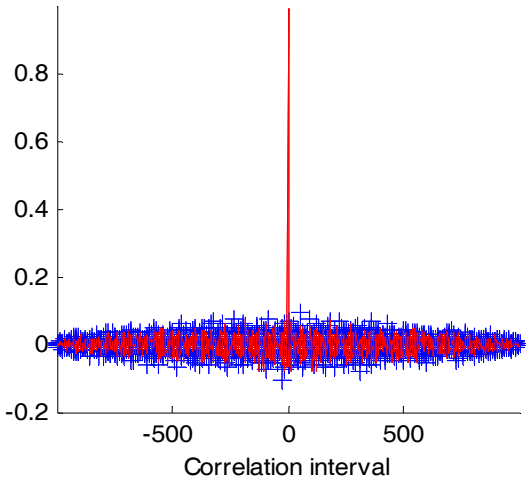


Figure 2.35: Fonctions d'auto et d'inter-corrélation.

La figure 2.36, montre que les séquences générées passent les tests de NIST (NIST, 2001a), affirmant ainsi, le caractère aléatoire de ses séquences.

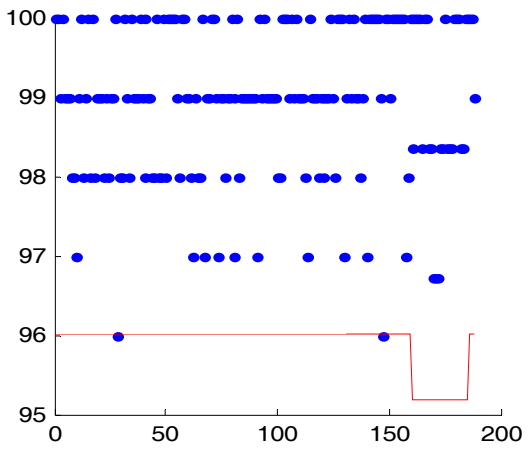


Figure 2.36: Résultats des tests de NIST.

Tous ces résultats, nous permettent d'affirmer que les séquences pseudo-chaotiques du générateur en question, peuvent être utilisées comme clés secrètes pour les algorithmes de chiffrement et d'authentification d'un système de communication sécurisé.

2.8.2. Simulation des performances du système de gestion des clés proposé

Dans ce paragraphe, nous calculons la valeur du *DO*, pour des flux des données IP utilisés dans des applications réelles. Les paquets ont été capturés avec Wireshark 1.0.6, et la liaison satellitaire sécurisée a été simulée avec Matlab. La figure 2.37, montre l'architecture du réseau simulé.

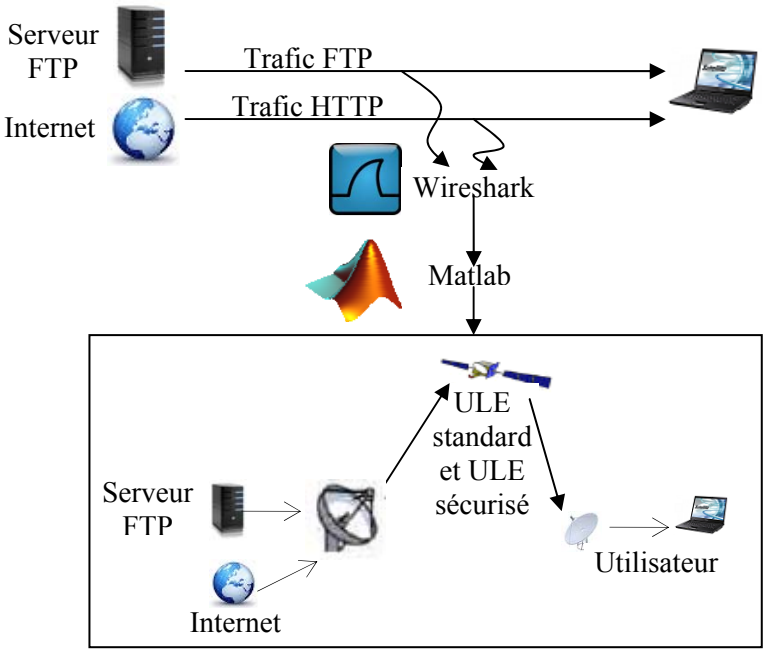


Figure 2.37: Architecture du réseau simulé

Nous avons réalisé deux liaisons IP: une liaison FTP serveur client et une liaison HTTP entre un serveur proxy, avec accès à Internet, et un client. Ceci nous a permis de couvrir deux types d'applications: le téléchargement des fichiers et la navigation Internet, pour les systèmes de communications. Tous les paquets IP ont été capturés avec Wireshark 1.0.6, et le trafic a été sauvegardé dans un fichier avec un format spécial, le format pcap (packet capture). Pour les communications FTP nous avons utilisé pour la simulation deux vitesses de transfert: 256 kbps et 90 Mbps. Ensuite, utilisant Matlab, nous avons extrait du fichier pcap, l'information relative à chaque paquet et nous avons simulé l'émission de tout le trafic IP par voie satellitaire.

Le *DO* est donné par la formule (2.5) comme le rapport entre la quantité d'information envoyée pour assurer la sécurité, *AI*, et la quantité totale d'information envoyée, *TI*. Alors, pour obtenir le *DO* nous devons connaître *AI* et *TI*. Pour cela, nous avons réalisé deux simulations: la première utilise l'encapsulation ULE standard, et la seconde utilise la solution de sécurité proposée. La simulation des communications IP par satellite avec l'encapsulation ULE standard nous a permis de quantifier l'*UUI* (Unsecured ULE Information) et la simulation de la solution de sécurité proposée nous a permis de quantifier le *TI* (Total Information). Les deux composantes : *UUI* et *TI* nous permettent de calculer *AI* selon la formule (2.13) :

$$AI = TI - UUI \tag{2.13}$$

Ainsi, nous avons utilisé la formule (2.13) et la formule (2.5) pour calculer le *DO* selon :

$$DO = \frac{AI}{TI} \cdot 100 = \frac{TI - UUI}{TI} \cdot 100 \qquad (2.14)$$

Les résultats obtenus sont montrés dans le tableau 2.3 :

Tableau 2.3 - DO total obtenu

	I	II	III	IV	V
FTP 90 Mbps	0.562	0.507	0.479	0.465	0.458
FTP 256 kbps	0.776	0.719	0.692	0.678	0.671
http	0.882	0.826	0.798	0.784	0.777

Comme nous avons accès à l'information au niveau de l'encapsulation ULE, nous avons pu identifier et quantifier l'information envoyée dans l'extension de l'en-tête (*EHI*) et l'information de gestion de clé (*KMI*). Ceci nous a permis de traiter aussi DO_{EH} et DO_{KM} selon les formules (2.8) et (2.7) respectivement. Ces résultats sont montrés dans les tableaux 2.4. et 2.5.

Tableau 2.4: DO_{EH} obtenu par simulation.

	FTP 90 Mbps	FTP 256 Kbps	HTTP
Taux des données ajoutées	0.454	0.668	0.773

Tableau 2.5: DO_{KM} obtenu par simulation.

	I	II	III	IV	V
FTP 90 Mbps	0.110	0.055	0.028	0.014	0.007
FTP 256 Kbps	0.112	0.056	0.028	0.014	0.007
HTTP	0.114	0.057	0.028	0.014	0.007

La simulation du système confirme les résultats théoriques obtenus. Nous ne pouvons pas simuler la période d'utilisation de *MK*, car le temps nécessaire pour sa simulation est énorme. Par exemple, pour un débit de 256 kbps et le jeu de paramètres V, la valeur théorique du *PMK* est de quelques dizaines d'années.

2.9. Conclusions et perspectives

Dans ce chapitre, nous avons proposé une nouvelle solution de sécurité pour les communications IP par DVB-S utilisant l'encapsulation ULE. Cette solution emploie une gestion multicouche des clés et utilise des fonctions chaotiques pour la génération des clés et pour le

chiffrement des données. Les services de sécurité offerts par cette solution sont la confidentialité, l'intégrité, et l'authenticité des données et la protection contre les attaques « envoie multiple » (*replay attacks*). Pour plus de sécurité, chaque PDU est chiffré et authentifié avec une clé différente. Aussi, la solution inclut un mécanisme qui permet le rétablissement de la synchronisation.

L'analyse et les résultats obtenus de cette solution de sécurité ont montré l'intérêt de la solution proposée en termes de taux des données ajoutées et aussi en termes de compatibilité entre les communications unicast et multicast.

En perspectives de ce travail, nous proposons de simuler la solution de sécurité proposée avec un simulateur DVB et d'étudier de manière plus approfondie le comportement de notre solution dans un environnement de communications multicast.

3. Amélioration de la sécurité de l'UMTS

3. Amélioration de la sécurité de l'UMTS

Dans ce chapitre nous étudions la sécurité des communications mobiles de troisième génération l'UMTS (Universal Mobile Telecommunications System). Nous présentons dans la première partie, la philosophie et l'architecture de la sécurité UMTS telle qu'elles sont vues par l'organisme de standardisation de la norme UMTS. Dans la deuxième partie de ce chapitre nous montrons des faiblesses identifiées dans la littérature spécialisée, des faiblesses nouvelles que nous avons identifiées nous-mêmes et nous proposons un nombre d'améliorations pour renforcer la sécurité des communications UMTS.

3.1. Contexte

La téléphonie mobile a connu un développement impressionnant ces dernières décennies. La téléphonie mobile, dite de deuxième génération, a connu un grand développement depuis son introduction commerciale au début des années 1990. Les services offerts se sont multipliés, les terminaux se sont améliorés, les protocoles sont devenus plus complexes et sécurisés, la structure du réseau est changée. Toutes ces modifications sont issues du progrès technique incluant les terminaux et du besoin d'interopérabilité globale des standards. Les technologies utilisées par la deuxième génération se basaient sur des normes issues des années 1980 comme : l'TTC au Japon (Telecommunications Technology Committee) et l'ETSIE (European Telecommunication Standards Institute - Europe). Différentes normes ont été implémentés, dont les plus importantes sont:

- GSM : (Global System for Mobile Communications) qui est le système le plus répandu. Il a été conçu en Europe mais il est utilisé par plus de 80% des abonnés au niveau mondial (www.wirelessintelligence.com).
- IS-95 : (Interim Standard 95) connu comme cdmaOne, ou CDMA qui a été conçu au Etats Unis et est utilisé sur les continents américains, en Asie et en Europe. Environ 17% des abonnés l'utilisent.
- PDC : (Personal Digital Cellular) est utilisé exclusivement au Japon.

Connaître les principes et la structure du GSM est très important, car le réseau UMTS garde les mêmes principes que le GSM, et sa structure, sécurité comprise, a été conçu comme une amélioration de la structure GSM, et non pas comme quelque chose de complètement nouveau. Pour cela nous présentons dans l'annexe B le fonctionnement et la structure des réseaux GSM et UMTS.

L'organisme en charge du standard UMTS est le 3 rd GPP (Generation Partnership Project). Il a été établie en décembre 1998 et est composé par l'ETSI (European Telecommunications Standards Institute), l'ARIB/TTC au Japan (Association of Radio Industries and Businesses/Telecommunication Technology Committee), l'CCSA (China Communications Standards Association), l'ATIS au Etats

Unis (Alliance for Telecommunications Industry Solutions) et l'TTA au Corée de Sud (Telecommunications Technology Association).

La sécurité des communications UMTS est prise en charge par l'organisation ETSI par son group expert en sécurité, ETSI SAGE (ETSI Security Algorithms Experts Group).

3.2. Philosophie de la sécurité dans l'UMTS

3.2.1. Principes de la sécurité 3G

Trois grands principes de sécurité ont été pris en compte quand les protocoles de sécurité 3G ont été développés : garder, le plus possible, les principes de la sécurité 2G; améliorer la sécurité 2G et offrir de nouveaux services. En ce qui suit, nous allons discuter chacun de ces principes (3GPP TS 33.120, 3GPP TR 33.900).

Même si la sécurité UMTS; est nettement meilleure que la sécurité GSM, ils restent toujours des aspects à améliorer.

3.2.1.1. Eléments de la sécurité 2G à retenir

Les éléments de la sécurité 2G qui doivent être retenus sont:

- authentification des abonnés pour avoir accès aux services;
- chiffrement des données envoyées sur la voie radio;
- protection de l'identité de l'abonné;
- utilisation de la carte USIM (Universal Subscriber Identity Module) comme module de sécurité ;
- possibilité de développement des applications de sécurité, implémentées sur l'USIM et qui communiquent directement avec le réseau d'origine ;
- procédures de sécurité incluse; c'est-à-dire, que l'utilisateur ne doit rien faire pour bénéficier des services de sécurité ;
- confiance dans le réseau de service doit être limitée.

3.2.1.2. Faiblesses de la sécurité 2G

Les faiblesses de la sécurité 2G qui doivent être corrigées sont:

- la possibilité de monter des attaques actives avec des fausses stations de base;
- les clés de chiffrement et les données d'authentification sont transmises en clair entre différentes éléments des réseaux;
- le chiffrement/déchiffrement est réalisé par le BTS (Base Transceiver Station), et les données sont transmises en clair entre le BTS et le BSC (Base Station Controller);

- l'authentification des données est réalisée par le chiffrement, et non avec une fonction spéciale. Il y a un grand risque de fraudes dans les réseaux qui n'utilisent pas le chiffrement;
- l'intégrité des données n'est pas assurée;
- le réseau d'origine ne contrôle pas l'utilisation des vecteurs d'authentification AV (Authentication Vector) par le réseau de service; en plus, il n'a aucune information à leur propos.

3.2.1.3. Nouvelles fonctions de la sécurité et les objectives de la sécurité

La sécurité de l'UMTS doit tenir compte de la diversité de fournisseurs des services et des services offerts et du fait que les services de la voix sont moins importants que les services de données. Ceci implique un risque accru d'attaques actives surtout lors de l'utilisation des services très sensibles tels que le commerce électronique.

En conclusion, les objectifs de la sécurité sont:

- la protection de l'information générée par ou relative à un utilisateur;
- la protection des services offerts par le réseau;
- l'interopérabilité des services à l'échelle mondiale;
- le niveau de protection offert doit être plus grand que dans le cas des systèmes antérieurs;
- la possibilité d'amélioration des procédures de sécurité pour combattre des nouvelles menaces ou pour protéger des nouveaux services.

3.2.2. Attaques et menaces principales

Après avoir présenté les principes et les objectives de la sécurité UMTS, présentons maintenant, les différentes menaces (A. Bais, 2006, Khan 2008, M. Hassan, 2010).

3.2.2.1 Le déni de service

Trois modalités d'attaque de type déni de service ont été identifié:

- *Usurpation des messages de déconnexion*: si le réseau ne peut pas authentifier les messages qu'il reçoit, un attaquant peut envoyer un message de *déconnexion* pour un autre mobile, qui a pour effet de couper la connexion entre le mobile en question et le réseau.
- *Usurpation des messages de changement de localisation*: cette modalité a un fonctionnement similaire avec l'attaque par usurpation des messages de déconnexion, c'est-à-dire un attaquant envoie un faux message de changement de localisation, ce qui a pour conséquence de couper la connexion entre le mobile et le réseau.
- C*onnexion avec des faux Nœuds B*: le mobile peut se connecter à un faux nœud *B*, contrôlé par un attaquant.

Ces attaques sont évitées par l'utilisation de la fonction intégrité pour les messages de contrôle et par la procédure d'authentification et d'établissement des clés (Authentication and Key Agreement – AKA) qui permet l'authentification d'accès aux réseaux.

3.2.2.2. Vol d'identité (Identity catching)

Le vol d'identité peut être réalisé de façon passive ou active. Si l'attaquant attend que l'utilisateur envoie son identité pour qu'il l'intercepte, il s'agit d'un vol passif d'identité. La protection contre ce type d'attaque est réalisée par l'utilisation d'identités temporaires.

Si l'attaquant utilise une fausse station de base pour demander aux utilisateurs qui s'y connecte leurs identités en clair, il s'agit d'un vol actif d'identité. La protection contre ce type d'attaque est réalisée par la procédure AKA qui rend impossible l'utilisation des fausses stations de base.

3.2.2.3. Se faire passer pour le réseau ou pour un abonné

Ces types d'attaques, supposent que l'attaquant peut utiliser une fausse station de base pour établir des connexions avec des utilisateurs ou que l'attaquant peut se connecter au réseau avec une fausse identité.

Si l'attaquant utilise une station de base, il peut alors réaliser le déchiffrement et accéder aux messages clairs envoyés par l'utilisateur. Si l'attaquant utilise une fausse identité pour se connecter au réseau, il peut accéder aux services de réseau de manière illégitime. Ces attaques sont rendues impossibles par l'utilisation de la procédure AKA permettant d'assurer l'intégrité des messages de contrôle et d'utiliser des nombres de séquences (SQN – Sequence Number).

Il faut mentionner aussi, qu'en pratique la procédure AKA peut être contournée dans le cas des réseaux qui supportent en plus le standard GSM (Meyer en 2004). Nous nous intéressons, dans notre travaille, seulement aux cas des réseaux UMTS purs.

3.2.2.4. Attaques cryptographiques

En général, la cryptanalyse intègre l'ensemble des moyens permettant de déchiffrer un texte crypté sans connaissance de la clé secrète. Il y a plusieurs types d'attaque standard, selon les quatre niveaux de complexité décroissante suivante:

- Attaque à texte chiffré seulement (ciphertext-only attack) – l'attaquant a connaissance seulement du texte chiffré de plusieurs messages (attaque la plus difficile). C'est celle à laquelle se trouvent régulièrement confrontés tous les services secrets.
- Attaque à texte en clair connu (known-plaintext attack) - l'attaquant détient plusieurs textes chiffrés ainsi que les textes en clairs correspondants, mais ce n'est pas lui qui a choisi les textes en clair.

- Attaque à texte en clair choisi (chosen plaintext attack) – l'attaquant peut choisir les textes en clair à chiffrer, mais il n'a pas le droit de modifier les textes en clair successifs proposés au système en fonction du résultat du chiffrement.

- Attaque à texte en clair choisi adaptative (adaptive-chosen plaintext attack) – c'est l'attaque la plus facile à mettre en œuvre. En effet, l'attaquant peut choisir les textes en clair qu'il donne à chiffrer au système et il peut les adapter (modifier) en fonction du résultat du chiffrement. Ceci permet au cryptanalyste de modifier son texte en clair en fonction du résultat du chiffrement correspondant et d'arriver ainsi assez vite à déchiffrer tout texte, si l'algorithme de chiffrement utilisé n'est pas assez sécurisé.

Notons enfin, que la sécurité dépend totalement de la clé secrète, et donc il est absolument nécessaire de la choisir aussi complexe que possible.

3.3. Architecture de la sécurité

3.3.1. Présentation de l'architecture de la sécurité UMTS

Dans ce paragraphe nous allons présenter la conception globale de la sécurité en UMTS. Elle est composée de 5 catégories, que nous allons présenter ci-dessous. Chacune de ces catégories, contrecarrée certaines menaces, et réponds à certains objectifs de la sécurité:

- *Sécurité au niveau accès* (Network access security) est l'ensemble des fonctions de sécurité qui offrent aux utilisateurs un accès sécurisé aux services de l'UMTS. Cette catégorie de la sécurité protège la liaison radio entre l'utilisateur et le reste du réseau UMTS.

- *Sécurité du domaine réseau* (Network domain security) est l'ensemble des fonctions de sécurité qui offrent aux nœuds du fournisseur un échange sécurisé des données de signalisation et une protection contre les attaques du réseau (wireline attacks).

- *Sécurité du domaine utilisateur* (User domain security) est l'ensemble des services de sécurité qui s'appuient sur l'accès des utilisateurs au téléphone mobile.

- *Sécurité du domaine application (*Application domain security) est l'ensemble des fonctions de sécurité qui permettent aux applications des utilisateurs, des échanges sécurisés des données avec le réseau.

- *Visibilité et configuration de la sécurité* (Visibility and configurability of security) est l'ensemble des fonctions de sécurité qui permettent à l'utilisateur de s'informer si une certaine fonction de sécurité est utilisée et si son accès aux services du fournisseur dépend de l'utilisation des fonctions de sécurité.

3.3.2. Sécurité au niveau accès réseau

Les spécifications techniques de 3GPP définissent cinq aspects de la sécurité au niveau accès réseau : confidentialité de l'identité de l'usager, authentification mutuelle usager-réseau, confidentialité des données, intégrité des données et identification de l'équipement mobile. Nous allons discuter chacun de ces aspects.

3.3.2.1. Confidentialité de l'identité de l'usager

Le standard UMTS affirme que l'architecture de la sécurité offre les services suivants, relatifs à la confidentialité de l'utilisateur:

- *Confidentialité de l'identité de l'utilisateur*: l'identité de l'utilisateur ne peut pas être dévoilée par écoute sur le canal radio ;
- *Confidentialité de la localisation de l'utilisateur*: la position d'un utilisateur ne peut pas être localisée en écoutant la voie radio ;
- *Non-traçabilité de l'utilisateur:* l'identité de l'utilisateur, auquel sont adressés les services de la voie radio, ne peut pas être connue pour un intrus.

Pour offrir ces services, l'utilisateur est en général, identifié avec des identifiants temporaires, TMSI (Temporary Mobile Subscriber Identity), ou avec son identifiant permanent chiffré, IMSI (International Mobile Subscriber Identity). Pour éviter la traçabilité de l'utilisateur, l'utilisateur ne doit pas être identifié pour une longue période de temps avec la même identité temporaire ou chiffré. Il y a des réseaux qui changent le TMSI chaque fois que l'utilisateur est authentifié (Kambourakis, 2005). En plus, tous les messages de contrôle qui peuvent divulguer l'identité de l'utilisateur doivent être chiffrés.

3.3.2.2. Authentification réciproque réseau - utilisateur

Le standard UMTS affirme offrir les services suivants, à propos de l'authentification réseau-utilisateur :

- *Authentification de l'utilisateur:* la propriété que le réseau peut vérifier l'identité de l'utilisateur.
- *Authentification du réseau*: la propriété que l'utilisateur peut vérifier l'identité du réseau.

Deux mécanismes d'authentification sont disponibles: un mécanisme avec des vecteurs d'authentification fournis par le réseau d'origine, plus précisément, le centre d'authentification (Authentication Center – AuC) et un mécanisme qui utilise la clé d'intégrité établie avec ce mécanisme.

3.3.2.3. Confidentialité et intégrité des données

Le standard UMTS affirme que l'architecture de la sécurité offre les services suivants:

- *Etablissement d'algorithmes d'intégrité et de confidentialité :* la propriété que l'UE (User Equipment) et le réseau de service peuvent négocier les algorithmes utilisés pour le chiffrement et l'intégrité des données.
- *Etablissement des clés secrètes de chiffrement et d'intégrité :* la propriété que l'UE et le réseau de service peuvent établir les clés secrètes utilisées pour le chiffrement et l'intégrité des données.
- *Confidentialité des données utilisateur et des messages de contrôle :* la propriété que les données utilisateur et les messages de contrôle ne peuvent pas être écoutés sur la voie radio par des entités non autorisées.
- *Intégrité et authentification des messages de contrôle :* la propriété que l'entité, qui reçoit les messages de contrôle peut vérifier s'ils ont été modifiés par un attaquant et qu'elle peut aussi vérifier l'identité de l'entité émettrice des messages.

3.3.3. Sécurité du domaine réseau

Le système de sécurité pour le domaine réseau, offre la sécurisation des messages envoyés sur le réseau IP et sur le réseau de signalisation pour les réseaux téléphoniques, SS7 (Signaling System no. 7).

Les réseaux IP sont sécurisés avec l'IPSec (IETF RFC 4301, 2005). Le standard UMTS utilise l'IPSec-ESP en mode tunnel permettant la sécurisation des messages IP entre les passerelles IP. La distribution et l'échange des clés sont réalisées avec l'IKE (Internet Key Exchange).

La sécurité sur le réseau SS7 est offerte par l'MAPSec (3GPP TS 33.200, 3GPP TS 29.002). Le protocole de sécurité pour le MAP (Mobile Application Part) est la partie du protocole SS7 spécifique pour la téléphonie mobile. Chaque message MAPSec consiste en un en-tête MAP et un corps de message protégé. Dans tous les cas, l'en-tête est envoyé en clair dans le réseau. MAPSec propose trois méthodes de protection :

- *Protection 0:* ce mode n'inclue aucune forme de protection.
- *Protection 1:* ce mode offre l'authentification et le contrôle d'intégrité des données MAP. Il utilise la fonction de sécurité f7.
- *Protection 2:* ce mode offre en plus, par rapport au mode *Protection 1*, la confidentialité des données en utilisant la fonction f6.

La gestion des clés du MAPSec est relativement lourde. Elle est gérée par un centre de gestion des clés, KAC (Key Administrator Center), dans chaque réseau. Les KACs communiquent entre eux à travers une interface IP et négocient les clés avec l'IKE.

L'architecture MAPSec permet un contrôle d'intégrité des messages, un contrôle de l'origine d'un message, une protection contre l'attaque par rejoue et une confidentialité des données.

3.3.4. Sécurité du domaine utilisateur

La sécurité du domaine utilisateur se porte sur deux aspects :

- *Authentification utilisateur – USIM :* cette authentification est réalisée avec un secret partagé par l'USIM (Universal Subscriber Identity Module) et l'utilisateur autorisé (ou un groupe d'utilisateurs autorisés), e.g. le PIN, (Personal Identification Number). Elle a pour but de limiter l'accès à l'USIM et donc aux services de l'UMTS, des utilisateurs non autorisés. Les mécanismes utilisés pour implémenter cette propriété sont décrits dans la norme 3GPP TS 31.101.
- *Authentification USIM – ME :* cette authentification est aussi réalisée avec des secrets partagés. Elle a pour but de limiter l'accès à l'ME (Mobile Equipment) pour les USIM qui ne sont pas autorisés. En pratique elle est utilisée par les opérateurs qui subventionnent le prix des mobiles afin de limiter l'utilisation de ces mobiles avec des USIM fournis par d'autres opérateurs.

3.3.5. Sécurité du domaine application

Cet aspect de la sécurité UMTS permet aux réseaux UMTS ou aux autres fournisseurs de services le développement des applications de sécurité sur la carte UICC (Universal Integrated Circuit Card). Ceci permet un transfert de données avec un degré de sécurité choisi par le réseau ou le fournisseur de l'application. Les spécifications techniques qui permettent le développement des applications sur l'UICC sont décrites par la norme 3GPP TS 31.111 USIM Application Toolkit. Les fonctions de sécurité pour ces applications sont décrites par la norme 3GPP TS 23.048 et elles répondent aux besoins de sécurité identifiés par l'TS 22.048.

3.3.6. Visibilité et configuration de la sécurité

Les spécifications techniques de l'UMTS donnent des suggestions vis-à-vis de la visibilité et de la possibilité de configuration des services de sécurité par l'utilisateur tels que:

- *Indication de chiffrement au niveau réseau d'accès:* la propriété que l'utilisateur est informé si le chiffrement est utilisé pour la communication avec le RNC (Radio Network Controller).
- *Annulation ou activation de l'authentification USIM – utilisateur:* la propriété que l'utilisateur est capable de contrôler si l'authentification USIM – utilisateur est activé ou non.
- *Acceptation ou rejet des appels non chiffrés:* l'utilisateur doit pouvoir contrôler s'il veut établir des appels quand le réseau n'a pas activé la procédure de chiffrement.

Pour cet aspect de la sécurité, le standard UMTS donne seulement des suggestions et laisse ainsi une grande liberté aux fournisseurs des services UMTS.

3.3.7. Vue d'ensemble de la sécurité de l'UMTS

Une vue d'ensemble de l'architecture de la sécurité UMTS est présentée dans la figure 3.1. La place de chacun des 5 aspects de la sécurité (présentés dans le paragraphe 3.3.1.) est montrée dans la structure du réseau UMTS (présentée dans l'annexe B). Les niveaux application, réseau et transport sont les niveaux du modèle OSI simplifié.

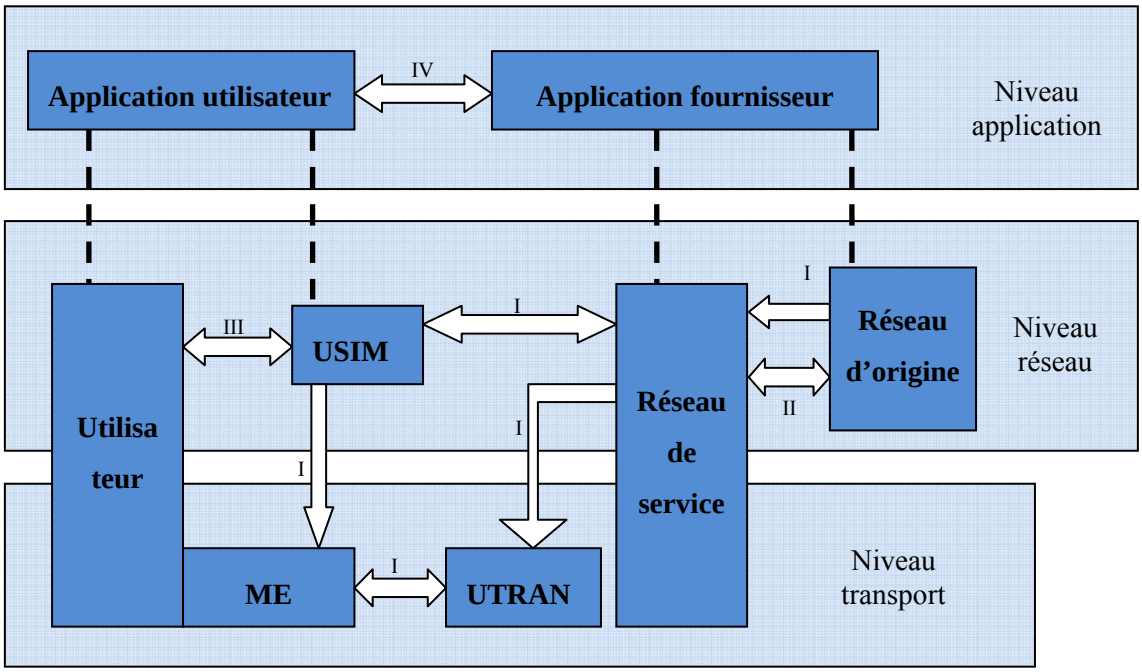


Figure 3.1: Architecture de la sécurité de l'UMTS.

C'est au domaine d'accès réseau que nous allons concentrer nos efforts, car c'est la partie la plus spécifique à l'UMTS et c'est le maillon faible de n'importe quel réseau de télécommunication sans fil.

La figure 3.2, montre la vue d'ensemble des principes d'enregistrement et de connectivité dans le cadre d'un réseau UMTS avec le domaine CS (Circuit Switched domain) et le domain PS (Packet Switched domain). L'identification, si c'est possible, est faite avec l'TMSI, sinon avec l'IMSI de l'utilisateur. L'authentification et l'établissement des clés seront faits de manière indépendante en chaque domaine. Les données et messages de contrôle seront chiffrés avec la clé de chiffrement CK_{CS} ou la clé CK_{PS} établie pour chaque domaine. Les messages de contrôle seront aussi authentifiés avec la clé d'intégrité IK_{CS} ou IK_{PS} établie pour chaque domaine.

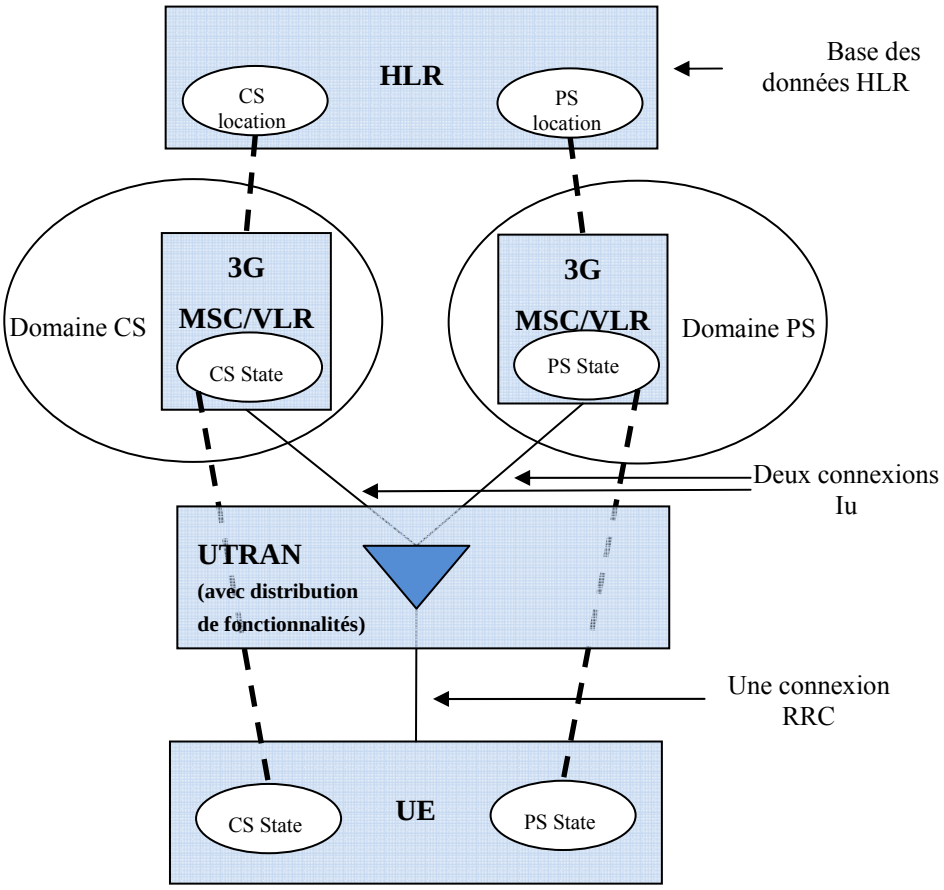


Figure 3.2: Vue d'ensemble sur les connexions de la sécurité dans un réseau UMTS

3.4. Accès sécurisé au réseau

3.4.1. Identification des utilisateurs

L'identification des utilisateurs peut être réalisée avec les identités temporaires TMSI ou avec l'identité permanente de l'utilisateur IMSI. L'identification temporaire TMSI et sa correspondance avec l'identification permanente IMSI sont retenues par le VLR/SGSN (Visitor Location Register/Serving GPRS Support Node) du réseau de service. Si un UE veut utiliser le TMSI avec un réseau de service différent, ou dans une zone différente, il doit aussi envoyer l'LAI (Location Area Identification) ou le RAI (Routing Area Identification) correspondant au TMSI utilisé pour que le nouveau réseau puisse l'identifier.

Le TMSI est utilisé seulement entre l'utilisateur et le réseau de service. Quand le réseau de service fait référence à l'utilisateur dans le cadre des communications avec le réseau d'origine, il utilise l'IMSI.

L'allocation d'un nouveau TMSI, TMSIn, est initiée par le VLR/SGSN. Le TMSIn est généré par le VLR et envoyé chiffrer à l'UE avec l'LAI. L'UE répond avec un message de confirmation. Une fois le changement est réalisé, l'UE efface l'ancienne valeur du TMSI, le TMSIa, et le VLR efface

l'association entre le TMSIa et l'IMSI de l'utilisateur (le TMSIa peut alors être utilisé par d'autres utilisateurs). Si cette procédure échoue, c.à.d que l'utilisateur ne reçoit pas le TMSIn, ou le VLR perd le TMSIa, alors, dans ces cas, l'utilisateur doit s'identifier auprès du VLR avec son identité permanente, l'IMSI.

Si l'utilisateur change sa localisation et entre dans une zone contrôlée par un autre VLR, VLRn, le VLR d'où il vient, VLRa, doit, normalement, transférer le TMSI de l'utilisateur au VLRn. Si ce transfert n'est pas possible, l'utilisateur doit s'identifier auprès du VLRn avec son identité permanente.

L'identification avec l'identité permanente, IMSI, est initiée par le VLR dans les deux cas décrits plus haut ou pour la première identification de l'utilisateur, lorsqu'il met en marche son portable. La réponse de l'utilisateur au message du VLR contient l'IMSI de l'utilisateur. Cette procédure est identifiée comme une faiblesse de sécurité de la norme UMTS. Dans le paragraphe 3.5.1, nous présentons notre proposition pour résoudre ce problème.

3.4.2. Authentification et établissement des clés AKA (Authentication and Key Agreement)

La procédure d'authentification et d'établissement des clés entre le réseau UMTS et le client est basée sur une clé secrète K, connue seulement par l'USIM et l'AuC (Authentication Center) du réseau d'origine. Cette procédure a été conçue comme une évolution de la procédure de sécurité du GSM pour offrir une meilleure compatibilité avec les équipements GSM, et de sa migration aisée vers l'UMTS. La procédure AKA est basée sur un ensemble de paramètres appelé vecteur d'authentification AV (Authentication Vector).

3.4.2.1. Vecteur d'authentification AV

Ce vecteur est généré par le réseau d'origine et il est envoyé au réseau de service dans le but de réaliser la procédure AKA. Six fonctions de sécurité: f0, f1,…f5 (que nous définissons dans le paragraphe 3.5.1) sont utilisées pour générer des vecteurs AV. Chaque vecteur AV est composé de cinq éléments:

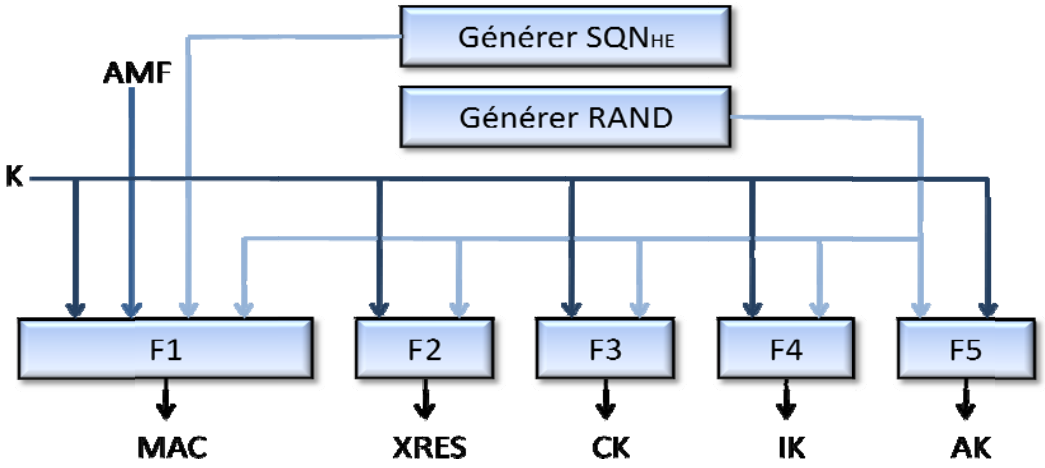
- RAND: une valeur aléatoire.
- XRES: la réponse attendue par le réseau.
- CK: la clé de chiffrement.
- IK: la clé d'intégrité.
- AUTN: un jeton d'authentification utilisé par le réseau.

Le paramètre RAND est une valeur aléatoire générée par la fonction f0. Il est envoyé en clair sur la voie radio et est utilisé comme entrée pour toutes les autres fonctions de sécurité f1 à f5.

Le réseau de service ne connaît pas la clé sécrète K, alors pour authentifier l'utilisateur, le réseau d'origine envoie XRES (généré par la fonction f2) au réseau de service. Ce paramètre ne peut pas être généré qu'avec la connaissance de la clé K. Si l'utilisateur envoie RES=XRES, l'authentification est réussie.

CK et *IK* sont les clés de chiffrement et d'intégrité générées respectivement par les fonctions f3 et f4. Le VLR reçoit ces clés du réseau d'origine. De son côté, l'USIM les génère au cours de la procédure AKA. Une fois l'authentification terminée, les clés seront envoyées aux équipements qui réalisent le chiffrement/déchiffrement, c'est-à-dire le RNC pour le réseau de service et le ME pour l'utilisateur. Le vecteur d'authentification AUTN est la concaténation de trois paramètres comme montre la figure 3.3.

La génération des éléments des vecteurs d'authentification AV est montrée par la figure 3.3. Ceci commence par la génération d'un nouveau numéro de séquence SQN_{HE} (Sequence Number Home Environment) et d'une nouvelle valeur aléatoire RAND. L'AuC utilise un compteur SQN_{HE} différent pour chaque utilisateur.



$$AV = RAND \| XRES \| CK \| IK \| AUTN$$

$$AUTN = SQN_{HE} \oplus AK \| AMF \| MAC$$

Figure 3.3: Génération des vecteurs d'authentification AV par l'AuC

La prédictibilité du SQN_{HE} peut compromettre l'identité des utilisateurs, pour cela, sa valeur est envoyée chiffrée avec la clé d'anonymat AK (Anonymity Key). L'USIM utilise une valeur seuil SQN_{MS} (Sequence Number Mobile Station) qui représente la valeur maximale du SQN_{HE} acceptée par l'USIM. Si la valeur SQN_{HE} reçue par l'USIM au cours de l'authentification est valide, l'authentification peut être réalisée avec succès. Sinon, l'USIM envoie une demande de resynchronisation.

Le champ de gestion de l'authenticité AMF (Authentication and key Management Field) est utilisé pour envoyer des informations relatives aux algorithmes utilisés pour modifier la valeur SQN_{MS} de l'USIM, ou pour déterminer la période de vie des clés CK et IK. Il peut être utilisé avec des

objectifs différents par des réseaux différents. Le standard UMTS offre seulement un cadre général pour l'utilisation du AMF.

La valeur MAC = $f1_K$(AMF, SQN_{HE}, RAND) est la partie du jeton AUTN qui sera vérifiée par l'USIM pour authentifier le réseau. L'utilisateur fait confiance à son réseau d'origine. Si le réseau de service peut fournir un bon message MAC, alors l'utilisateur saura que son réseau d'origine a fait confiance au réseau de service et, par conséquence, l'utilisateur aussi.

Le message AUTN est la concaténation des trois valeurs ($SQN_{HE} \oplus AK$), AMF, et MAC. Il est envoyé par le réseau de service à l'utilisateur, pour que ce dernier puisse authentifier le réseau de service et avoir des informations sur certains aspects de la sécurité, comme par exemple les algorithmes de chiffrement utilisés.

3.4.2.2 Procédure AKA

Le déroulement de la procédure d'authentification et d'établissement des clés est présenté dans la figure 3.4.

Figure 3.4: Déroulement de la procédure AKA

La procédure AKA est initiée par le réseau de service. Dans une première étape, le réseau de service demande au réseau d'origine les vecteurs d'authentification AV(1…n) qui forment une queue de type « premier arrivé, premier servi » (FIFO – First In First Out). Ensuite, le réseau de service choisit un de ces vecteurs, AV(i), et l'utilise pour l'authentification. Un vecteur d'authentification peut être utilisé pour une seule authentification seulement. Le réseau de service envoie à l'USIM une requête d'authentification (*user authentication request)* qui contient les parties RAND(i) et AUTN(i). L'USIM vérifie l'authenticité du vecteur AUTN(i) et, s'il est authentique, alors, l'USIM traite les clés CK et IK et produit la réponse RES(i), qui est envoyée au réseau de service. Le réseau de service

vérifie la réponse RES(i) et, si celle-ci est correcte, alors, le réseau de service sélectionne les clés CK et IK. Les procédures d'authentification réalisées par l'USIM sont montrées par la figure 3.5 :

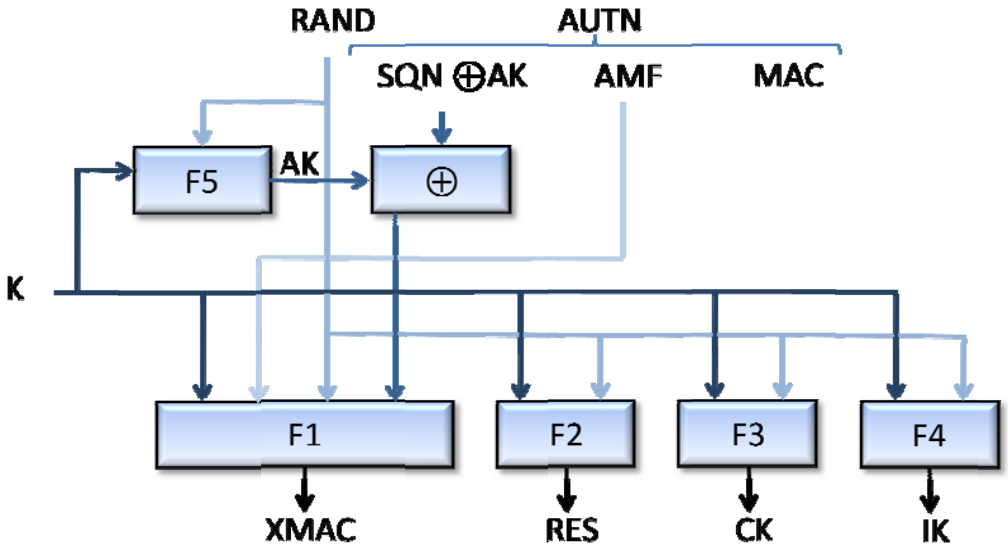


Figure 3.5: Fonctions d'authentification effectuées par l'USIM

Au début l'USIM utilise K et RAND pour l'obtention de la clé AK = $f5_K$(RAND) permettant le déchiffrement du numéro de la séquence SQN = (SQN **XOR** AK) **XOR** AK. Ensuite, l'USIM utilise la valeur aléatoire RAND et la clé K pour calculer la valeur XMAC = $f1_K$(RAND, SQN, AMF) et de vérifier si la valeur calculée est égale à la valeur reçue MAC, contenue dans le jeton d'authentification, AUTN. Si les valeurs en question sont différentes, l'utilisateur envoie un message d'échec d'authentification (*user authentication reject*) au réseau de service, message qui inclue la raison de l'échec. Le réseau de service, dans ce cas, envoie au HLR du réseau d'origine, un rapport avec l'IMSI de l'utilisateur sur: la cause d'échec, le type d'accès qui a initié la procédure d'authentification, les autres tentatives sans succès, l'adresse du VLR/SGSN et la valeur RAND du vecteur d'authentification.

Si les valeurs MAC et XMAC sont égales, alors l'USIM confirme la validité du numéro de la séquence, SQN_{HE}. Autrement, l'USIM envoie un message d'échec de la synchronisation (*synchronization failure*), AUTS qui va permettre au réseau de service de demander au réseau d'origine des nouveaux vecteurs d'authentification.

Si le SQN_{HE} est valide, l'USIM traite la réponse RES = $f2_K$(RAND) et inclue cette valeur dans le message réponse (*user authentication response*) qu'il envoie au réseau de service. Ensuite, il traite les clés de chiffrement et d'authenticité CK et IK et les envoie à l'ME qui va les utiliser pour le chiffrement et l'authenticité des données. La procédure d'authentification et d'établissement des clés est réalisée par l'USIM, mais le chiffrement et l'authenticité des messages sont faits par le l'ME.

Quand le VLR/SGSN reçoit le message *user authentication response*, il vérifie si la valeur RES contenue dans le message est égale à la valeur XRES du AV utilisée. Si ce n'est pas le cas, le réseau

de service envoie au réseau d'origine le même rapport d'échec envoyé lorsque le MAC et l'XMAC sont différents. Si le RES et l'XRES sont égaux, le VLR/SGSN sélectionne les clés CK et IK contenues par l'AV et les envoie au RNC. C'est le VLR/SGSN qui est responsable de la procédure AKA, mais le chiffrement et l'authentification des messages sont réalisés par le RNC.

La procédure AKA est réalisée de manière indépendante pour le domaine CS et pour le domaine PS, comme nous l'avons déjà décrit dans la figure 3.2 du paragraphe 3.3.1.

La longueur des paramètres utilisés au cours de la procédure AKA est présentée dans le tableau 3.1:

Tableau 3.1. : Longueur des paramètres AKA

Paramètre	Longueur (bits)
K	128
RAND	128
SQN	48
AK	48
AMF	16
MAC, XMAC	64
CK	128
IK	128
RES	32-128

3.4.3. Intégrité des données et confidentialité

Une fois la procédure AKA est accomplie, la fonction de protection de l'intégrité est utilisée pour l'authentification locale entre le terminal et le réseau de service. Cette protection utilise les clés établies avec la procédure AKA au lieu d'utiliser les AV fournis par le réseau d'origine.

3.4.3.1. Négociation des algorithmes

Quand l'UE veut établir une connexion avec le réseau UMTS, il envoie un message indiquant les algorithmes de chiffrement et d'intégrité supportés. Le réseau les compare avec ses propres algorithmes d'authentification UIA (UMTS Integrity Algorithm), ses préférences et les éventuels requis spéciaux de l'abonnement, puis suivra les règles suivantes:

1. Si l'UE et le réseau n'ont aucun UIA en commun, la connexion sera libérée.
2. S'ils ont au moins un UIA en commun, alors le réseau choisira un de ces algorithmes pour l'utiliser dans le cadre de cette connexion.

Le réseau comparera ses algorithmes de chiffrement UEA (UMTS Encryption Algorithm), préférences et les éventuels requis spéciaux de l'abonnement avec ceux supportés par l'UE et suivra les règles suivantes :

1. Si l'UE et le réseau n'ont aucun UEA en commun et que le réseau n'est pas prêt à utiliser une connexion non chiffrée, la connexion sera abandonnée.
2. Si l'UE et le réseau n'ont aucun UEA en commun et que l'utilisateur et le réseau sont prêts à utiliser une connexion non-chiffrée, alors ils vont établir une connexion non-chiffrée.
3. S'ils ont au moins un UEA en commun, alors le réseau choisira un de ces algorithmes pour l'utiliser dans le cadre de cette connexion.

Même si la procédure AKA se déroule indépendamment pour les deux domaines, PS et CS, les algorithmes UIA et UEA négociés pour un de ces domaines ne peuvent pas être changés pour l'autre. Alors, les mêmes algorithmes UEA et UIA sont utilisés pour les deux domaines CS et PS.

3.4.3.2. Validité des clés

Un mécanisme qui permet le contrôle de la période de validité des clés est nécessaire afin d'assurer une protection contre les attaques avec des clés compromises. Ceci est possible car la procédure AKA n'est pas utilisée chaque fois qu'un appel est initié.

Chaque fois qu'une connexion RRC (Radio Resource Control) est terminée, les valeurs des compteurs $START_{PS}$ ou $START_{CS}$, envoyées par le réseau, sont comparées par l'USIM avec leur valeur maximale acceptée, THRESHOLD. Si les valeurs en question sont plus petites que la valeur maximale, leur valeur actuelle est stockée par l'USIM et les clés CK et IK actuelles restent valables. Sinon, l'UE va marquer la valeur actuelle du $START_{CS}$ ou $START_{PS}$ comme invalide, effacera les clés actuelles CK et IK et mettra le paramètre KSI (Key Set Identifier) à '111', ce qui indique que les clés ne sont plus valides.

Quand une nouvelle connexion RRC sera ouverte, la validité de la valeur START sera vérifiée. Si la valeur START est marquée comme invalide (c.à.d égale ou supérieure à THRESHOLD) l'UE déclenchera une nouvelle procédure AKA.

3.4.3.3. Etablissement de la sécurité

Nous allons décrire le début des procédures de chiffrement et d'intégrité. La protection de l'intégrité pour les messages de contrôle est nécessaire pour chaque établissement de connexion entre

l'ME et le VLR/SGSN. Les exceptions à cette règle ne sont pas nombreuses et incluent les appels aux services d'urgence et les messages de contrôle qui confirment l'état actuel sans le changement.

Les seules opérations permises, après l'envoie du « message L3 initial » (message envoyé quand on met en marche le téléphone mobile) et avant l'établissement d'une connexion sécurisée, sont la procédure AKA et l'identification avec une identité permanente.

La figure 3.6, montre les étapes qui mènent à l'établissement d'une connexion sécurisée entre l'utilisateur et le réseau :

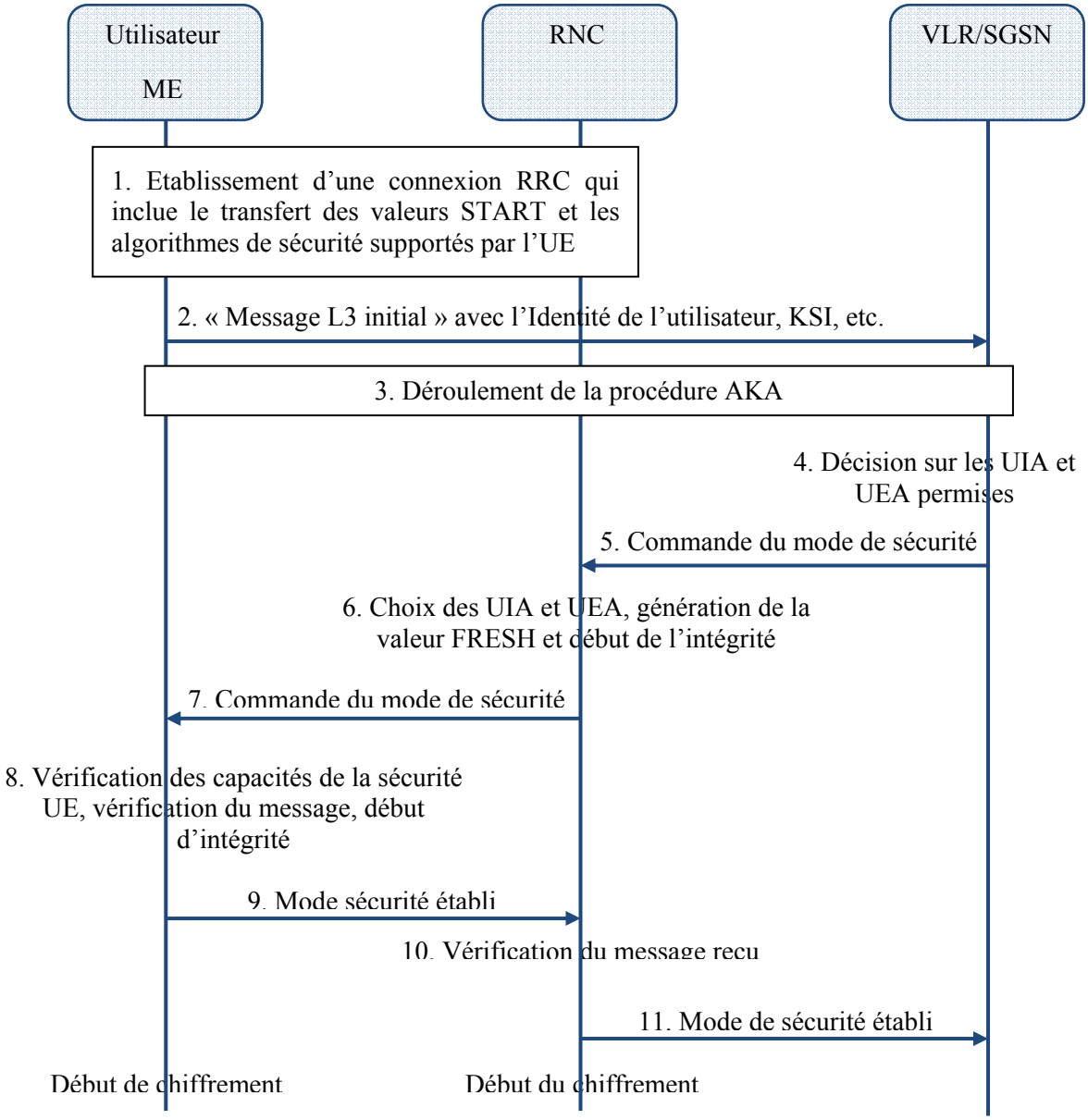


Figure 3.6: Authentification locale et établissement de la connexion

Les étapes de la figure 3.6 sont:

1. Etablissement d'une connexion RRC, incluant la transmission vers le RNC des valeurs START pour les deux domaines ($START_{PS}$ et $START_{CS}$) et les algorithmes de sécurité

supportés par l'UE : les UIA et UEA. Ces informations sont stockées par le RNC qui va les utiliser plus tard.

2. L'utilisateur envoie le « message L3 initial » qui contient aussi l'identité permanente de l'utilisateur et le KSI. Les deux premières étapes sont réalisées seulement quand le mobile est allumé. Une fois ces informations transmises, la procédure AKA peut être initiée plusieurs fois et la procédure va commencer avec l'étape 3.
3. Si le KSI est égal à '111', il n'y a pas un ensemble de clés valides et la procédure AKA est lancée entre l'UE et le réseau de service (VLR/SGSN). Sinon, il y a une connexion sécurisé établie et les clés CK et IK seront utilisées.
4. Le VLR/SGSN décide les UEA et UIA autorisés et l'ordre de préférence.
5. Le VLR/SGSN utilise le message RANAP « commande du mode de sécurité » (security mode command) pour initier l'intégrité et le chiffrement. Ce message contient une liste ordonnée des algorithmes de sécurité (voir 4.) et les clés IK et CK. Le message indique aussi si les clés CK et IK viennent d'être établies (voir 3.) ou si elles ont été établies avant et ont déjà été utilisées. Si les clés ont été utilisées avant, la valeur du START envoyée au point 1 est utilisée. Sinon, elle est mise à zéro.
6. Le RNC décide les algorithmes qui seront utilisés. Il choisit le plus performant des algorithmes en fonction de la liste ordonnée reçue au point 5 et des algorithmes de sécurité supportés par l'UE, reçu au point 1.
7. Le RNC génère le message RRC « commande du mode de sécurité » avec les algorithmes qui seront utilisés, la valeur FRESH (voir 3.3.4.4.), la commande de début de chiffrement, le domaine (CS ou PS) et le MAC d'intégrité de message. Ce message représente le début de l'intégrité pour la liaison descendante.
8. L'UE vérifie le message RRC « commande du mode de sécurité » reçu et commence la procédure d'intégrité pour la voie montante.
9. Si le message est valide, l'UE envoie le message de confirmation « mode sécurité établi » et lui attache le code MAC. Sinon, la procédure s'arrête là.
10. Le RNC vérifie l'intégrité du message reçu.
11. Si le message reçu est valide, le RNC envoie au VLR/SGSN le message RANAP « mode de sécurité établi » qui inclue les algorithmes sélectionnés.

Les premières implémentations des UEA dans les UE ont été défectueuses : il y a des attaques qui exploitent ces faiblesses et arrivent à accéder à l'information chiffrée. Pour cela le VLR/ SGSN peut établir l'algorithme à utiliser en fonction du modèle du terminal utilisé. Le modèle du terminal est contenu dans le message d'identité internationale de l'équipement IMEI (International Mobile Equipment Identity).

S'il s'agit d'un UE avec une implémentation défectueuse d'UEA1, alors le VLR/SGSN prendra la décision de ne pas utiliser le chiffrement (utiliser UEA0). Dans ce cas, une fois la connexion de sécurité établie, le SGSN va demander encore une fois l'IMEI, cette fois avec la protection d'intégrité activée. Si l'IMEI reçu est différent de l'IMEI reçu avant l'établissement d'une connexion sécurisée, la liaison sera libérée. Sinon, la liaison sera établie.

3.4.3.4. Protection de l'intégrité

La protection de l'intégrité est utilisée pour la plupart des messages de contrôle et elle n'est pas utilisée pour la protection des données. Cette protection est réalisée par un code MAC qui est ajouté à la fin des messages. A cet effet, la fonction de sécurité f9 est utilisée. La figure 3.7, montre la manière d'utilisation de cette fonction. Les entrées de la fonction f9 sont :

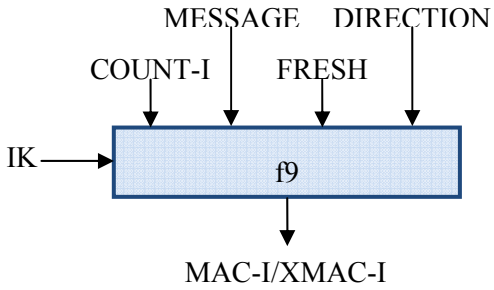


Figure 3.7: Protection de l'intégrité des messages

- IK : la clé d'intégrité.
- COUNT-I: un compteur de séquences sur 32 bits.
- MESSAGE: est le message qui est protégé.
- FRESH: une valeur aléatoire établie par le réseau (voir 3.3.4.3, étape 7.) chaque fois qu'une liaison de sécurité est établie avec un UE. Ensuite elle est utilisée par le réseau et l'UE afin d'éviter l'utilisation des anciens codes MAC-I.
- DIRECTION: un bit qui indique le sens du message (voie montante ou descendante). Ce paramètre est utilisé pour éviter l'utilisation des mêmes paramètres d'entrée pour les deux voies.

Le côté qui envoie le message, calcule le code MAC-I et l'ajoute au message. Le récepteur effectue le calcul de XMAC-I et le compare avec le MAC-I reçu. S'ils sont différents, le message sera ignoré.

Algorithmes d'intégrité

Chaque algorithme d'intégrité est identifié par une valeur sur 4 bits. Seulement 2 valeurs ont été définies :

- '0001': UIA1, KASUMI, détaillés en 3.4.2.1.
- '0002 : UIA2, SNOW 3G, détaillés en 3.4.2.2.

Ces algorithmes sont présentés ultérieurement dans le paragraphe 3.5.2.

Les autres valeurs ne sont pas définies. L'UE et le RNC doivent implémenter ces deux algorithmes.

3.4.3.5. Protection de la confidentialité

La protection de la confidentialité est utilisée pour les messages de contrôle et pour les données aussi. Le chiffrement est réalisé avec la fonction de chiffrement f8. L'utilisation de la fonction f8 est montrée par la figure 3.8.

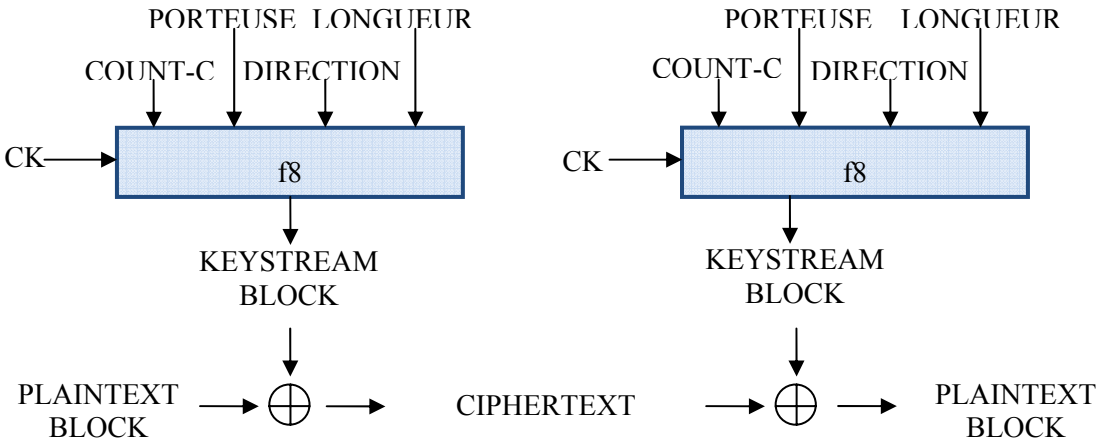


Figure 3.8: Protection de la confidentialité des messages

La confidentialité est obtenue par ou exclusive entre le message clair et le « keystream block ». De manière similaire, le déchiffrement est réalisé avec l'opération ou exclusive entre le texte reçu (qui est chiffré) et le même « keystream block » reconstruit en réception.

Les paramètres utilisés pour la fonction f8 sont :

- CK: clé de chiffrement.
- COUNT-C: compteur de séquences sur 32 bits pour le chiffrement.
- PORTEUSE: identifiant sur 5 bits de la porteuse. Il est utilisé pour éviter l'utilisation de la même fonction de chiffrement pour des porteuses différentes.
- LONGUEUR: valeur sur 16 bits qui indique la longueur nécessaire du « keystream block »
- DIRECTION: bit qui indique le sens du message (voie montante ou descendante). Ce paramètre est utilisé pour éviter l'utilisation des mêmes paramètres d'entrée pour les deux voies.

Algorithmes de chiffrement

Chaque algorithme de chiffrement est identifié par une valeur sur 4 bits. Seulement trois valeurs sont définies :

- '0000' : UEA0, aucune fonction.
- '0001' : UEA1, KASUMI, détaillés en 3.4.2.1.

- '0002' : UEA2, SNOW 3G, détaillés en 3.4.2.2.

L'UE et le RNC doivent implémenter ces trois algorithmes.

3.5. Fonctions de sécurité

Au total 9 fonctions de sécurité sont utilisées par l'UMTS. Ces fonctions ont été développées par l'ETSI SAGE (ETSI Security Algorithms Experts Group). Elles sont divisées en deux catégories: les fonctions utilisées durant la procédure AKA et les fonctions utilisées pour le chiffrement et l'intégrité des données.

3.5.1. Fonctions utilisées pour la procédure AKA

La procédure AKA utilise 7 fonctions de sécurité: f1, f1*, f2, f3, f4, f5 et f5* dont deux (f1* et f5*) pour la partie de resynchronisation, en cas de perte de synchronisation entre les clés du ME et les clés du réseau. Les spécifications 3GPP TS 35.205, 3GPP TS 35.206, 3GPP TS 35.207, 3GPP TS 35.208, 3GPP TS 35.909 donnent un exemple de jeu d'algorithmes qui peuvent être utilisés pour ces fonctions de sécurité et analysent leurs caractéristiques. Le standard UMTS laisse aux opérateurs le choix des algorithmes qui seront implémentés. Ce n'est pas obligatoire d'implémenter ce jeu d'algorithmes là, il est juste donné comme exemple de jeu d'algorithmes qui peuvent être utilisés.

La figure 3.9, montre la structure du jeu d'algorithmes proposés par le 3GPP pour les fonctions de sécurité f1 à f5. Dans cette figure, le block E_K représente un algorithme de chiffrement utilisant la clé K, r1 à r5 sont cinq constantes de rotation et c1 à c5 sont cinq constantes définies par le 3GPP TS 33.105.

Figure 3.9: Structure des fonctions de sécurité f1, f1, f2, f3, f4, f5, f5**

La taille de chaque lien et des sorties est de 128 bits. La fonction f1 a une taille de 64 bits (qui sont les bits de poids faible de la première sortie formée par concaténation de f1 et f1*). La fonction f1* a, aussi, une taille de 64 bits, qui sont les 64 bits de poids fort. De même, la fonction f5 est formée de 64 bits (dont seulement 48 sont utilisés pour AK) de poids faible de la deuxième sortie et f2 contient les 64 bits de poids fort de cette même sortie.

Pour chaque fonction, le traitement des données entre l'entrée et la sortie inclus deux utilisations d'un algorithme de chiffrement E_K, une rotation basée sur un nombre spécifique de bits ('Rotation ri bits'), une addition avec une constante spécifique (ci) et trois additions avec une constante OP_C. Cette complexité peut expliquer pourquoi ces fonctions sont encore considérés robustes, après plus de 10 ans d'utilisation.

Le noyau de ces fonctions est l'algorithme de chiffrement E_K. Les spécifications 3GPP proposent l'utilisation de l'algorithme de chiffrement par bloc Rijndael, devenu AES après une modification mineure. L'utilisation de l'algorithme AES pour le noyau des fonctions n'est pas obligatoire. Les opérateurs peuvent remplacer cet algorithme par un autre, s'ils considèrent que cela les conviennent mieux. Quand même, le noyau doit être un algorithme de chiffrement par bloc de taille de 128 bits et une clé de 128 bits. Cette spécification a été utilisée pour pouvoir offrir des fonctions de sécurité très robustes de point de vue cryptographique.

La valeur OP_C est calculée comme suit:

$$OP_C = OP \oplus E(OP)_K \tag{3.1}$$

où *OP* est une constante spécifique pour chaque operateur.

La norme 3GPP TS 35.206 recommande que la valeur OP_C soit calculée en dehors de la carte à puce et soit mémorisée par la suite sur la carte à puce comme valeur individuelle. Chaque operateur décide si la valeur *OP* est publique ou secrète.

3.5.2. Fonctions utilisées pour le chiffrement et l'intégrité

L'ETSI SAGE a développé deux jeux d'algorithmes de chiffrement et d'intégrité:

- UIA1 et UEA 1 qui utilisent l'algorithme de chiffrement par blocs Kasumi
- UIA2 et UEA2 qui utilisent l'algorithme de chiffrement par flux SNOW 3G.

3.5.2.1. UIA1 et UEA1 utilisant l'algorithme KASUMI

L'algorithme de chiffrement KASUMI constitue le noyau des algorithmes UIA1 et UEA1. C'est un algorithme de chiffrement par bloc de 64 bits utilisant 8 séries Feistel et une clé secrète de taille 128 bits. La figure 3.10, montre comment est utilisé l'algorithme de KASUMI pour calculer la fonction f9 assurant l'intégrité des messages de contrôle. Nous rappelons qu'uniquement les messages de contrôle ont l'intégrité protégée, les données sont seulement chiffrées.

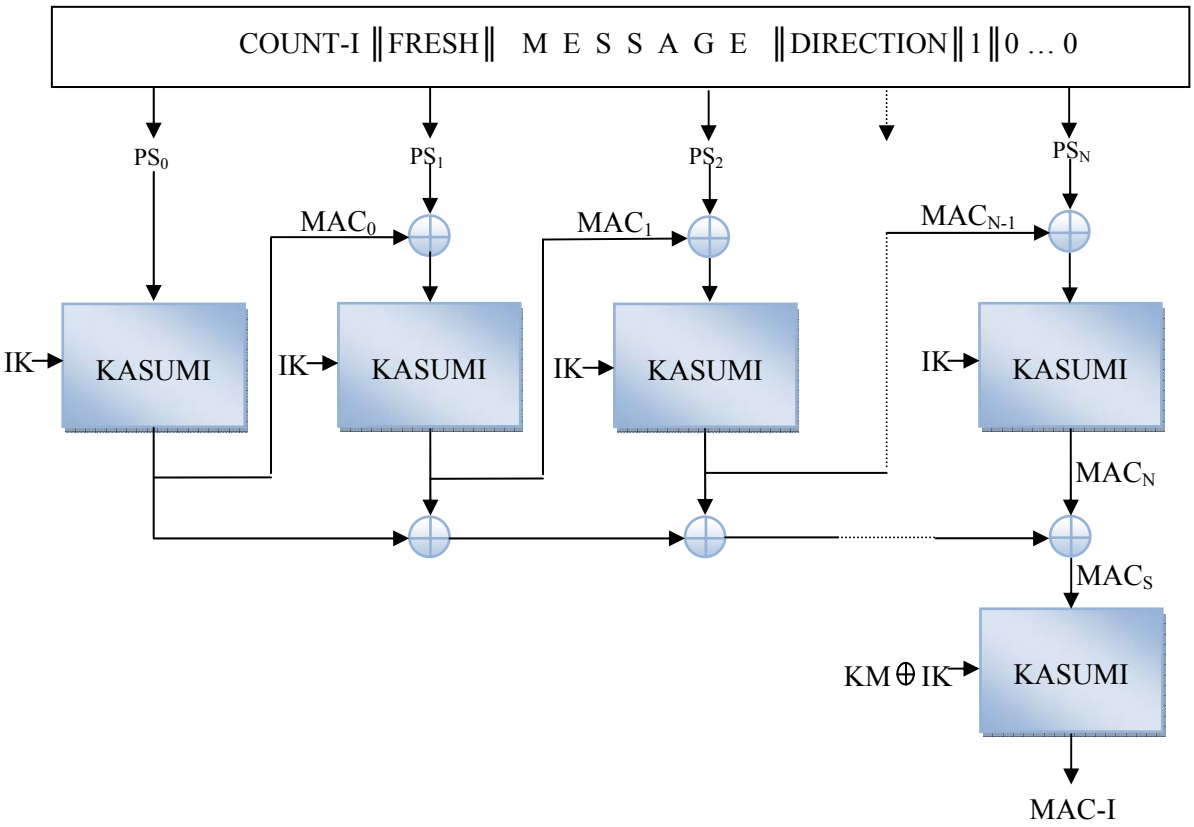


Figure 3.10: Fonction d'intégrité f9 utilisant l'algorithme KASUMI (UIA1)

Les données pour lesquelles on calcule le MAC (*COUNT-I ‖FRESH‖ M E S S A G E ‖DIRECTION‖1‖0 … 0*) sont divisées en blocs de 64 bits chacun, PS_i avec $i=0...N$. Le premier bloc chiffré avec l'algorithme de KASUMI est le bloc PS_0. Le résultat du chiffrement, noté MAC_0, est mémorisé. Ensuite, les différents blocs PS_i sont traités en mode chainage des blocs (mode cryptographique CBC). Tous les MAC_i avec $i=0...N$, mémorisés, sont additionnés modulo 64 et le résultat, noté MAC_S, est chiffré avec l'algorithme KASUMI. Mais cette fois ci, la clé secrète utilisée par l'algorithme n'est pas IK, mais IK+KM, où KM est une constante. Le résultat de ce chiffrement est le code MAC-I.

La figure 3.11, montre comment est réalisée la fonction de chiffrement f8, basée sur l'algorithme KASUMI pour le chiffrement des données.

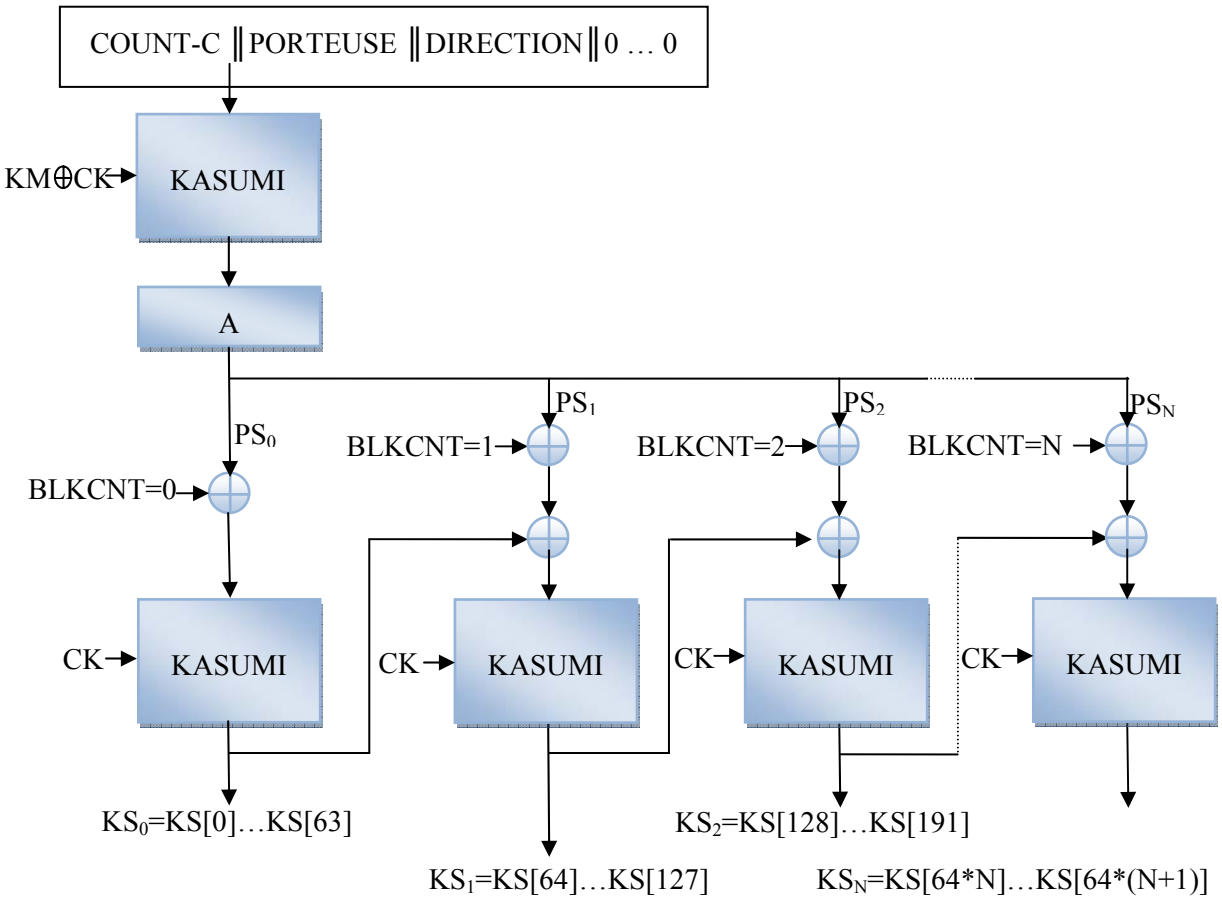


Figure 3.11: Fonction de chiffrement f8 utilisant l'algorithme KASUMI (UEA1)

Le bloc A est un registre à 64 bits utilisé pour mémoriser les valeurs intermédiaires, BLKCNT est un compteur sur 64 bits et KM est la même constante utilisée par la fonction f9 pour modifier la clé.

Les entrées de la fonction *f8* (COUNT-C ‖PORTEUSE ‖DIRECTION‖0 … 0) sont divisées en blocs de 64 bits chacun et sont chiffrées séquentiellement bloc par bloc par l'algorithme KASUMI. Les résultats intermédiaires sont mis dans le registre A (PS_i avec $i=0...N$), ensuite, ils sont additionnés, avec les compteurs $BLKCNT_i$ pour former une série d'entrées intermédiaires qui sont chiffrées par une batterie d'algorithmes de Kasumi en mode CBC. La sortie de la fonction de chiffrement f8 est le keystream KS.

L'algorithme KASUMI a été cassé en 2005 par Biham, Dunkelman et Keller. Cependant, la complexité de l'attaque rendrait impossible son utilisation pratique. Mais, en 2010 Dunkelman, Keller et Shamir ont publié une méthode de cryptanalyse facile à implémenter et permet de retrouver la clé de chiffrement. Alors les algorithmes UIA1 et UEA1 ne sont plus considérés comme des algorithmes robustes.

3.5.2.2. UIA2 et UEA2 utilisant l'algorithme SNOW 3G

L'algorithme SNOW 3G est un algorithme de chiffrement par flux utilisant une clé de 128 bits et une variable d'initialisation de 128 bits. Il est basé sur l'algorithme SNOW 2.0 et a subi des modifications pour devenir plus robuste contre la cryptanalyse algébrique (O. Billet, 2005) et contre les attaques de type distinguées « distinguishing attacks » (D. Watanabe , 2004). Conformément aux spécifications (ETSI/SAGE, 2006) l'algorithme SNOW 3G est composé d'un générateur pseudo-aléatoire de longueur maximale LFSR (Linear Feedback Shift Register) formé de 16 éléments dans le champ Galois GF(2^{32}) (chacun de ces 16 éléments contient 32 bits) et d'un automate à état fini FSM (Finite State Machine) composé de 3 états, chacun sur 32 bits. La sortie de ce système est un mot de 32 bits utilisé pour chiffrer l'information.

La figure 3.12 montre l'utilisation de l'algorithme SNOW 3G dans le cadre de la fonction d'intégrité f9 UIA2 :

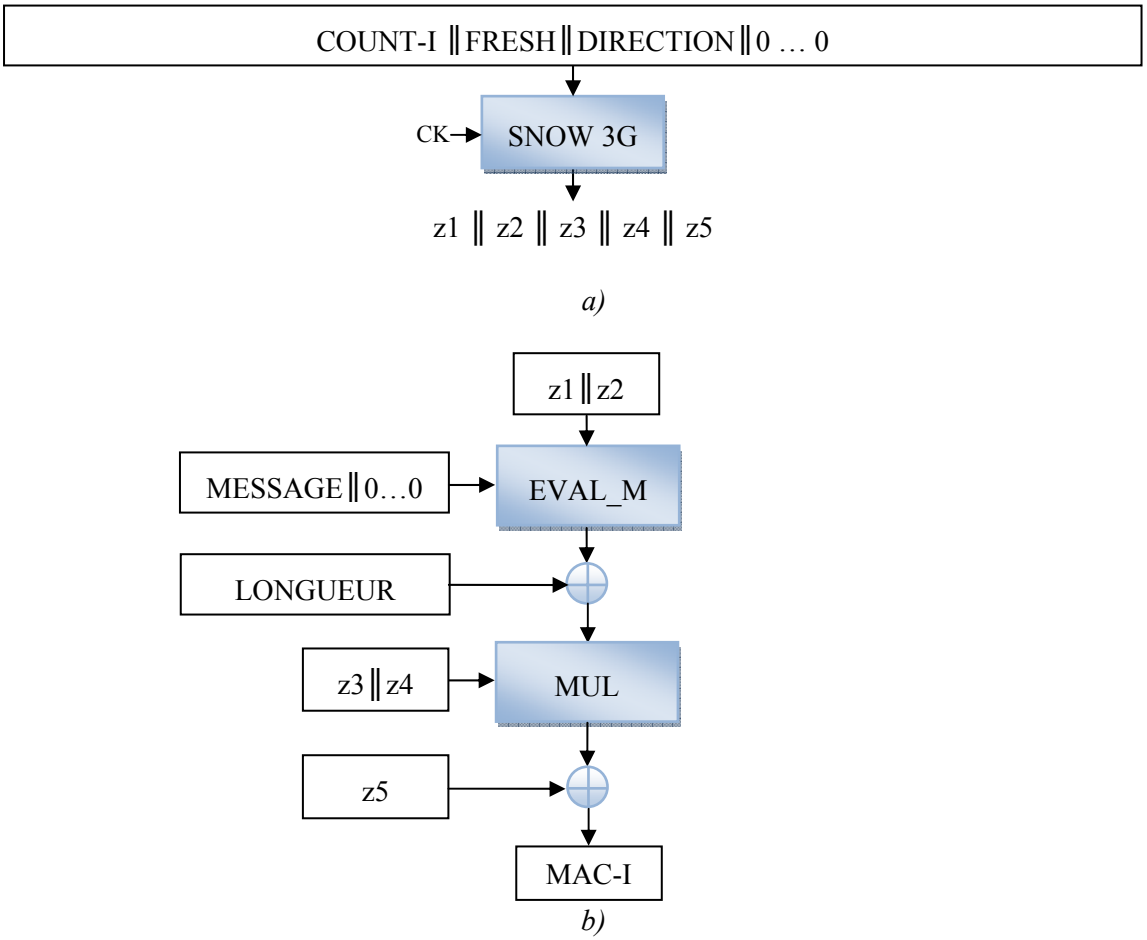


Figure 3.12. : Fonction d'intégrité f9 utilisant l'algorithme SNOW 3G(UIA2) a) première partie, b) deuxième partie

Dans la partie *a),* l'algorithme SNOW 3G utilise les entrées de la fonction *f9*, COUNT-I, FRESH et DIRECTION, comme variable d'initialisation et génère en sortie cinq valeurs, chacune sur 32 bits, *zi* avec *i=1…5*. Ces valeurs et le message sont utilisés dans la partie *b)* pour générer le code MAC. La fonction EVAL_M fait l'évaluation d'un polynôme dans le point P. Le polynôme est définie

sur l'espace Galois GF(2^{64}) et est composé du message et de sa longueur. Le point P est défini par z1‖z2. La fonction MUL fait la multiplication des polynômes d'entrée. Il s'agit toujours des opérations dans l'espace GF(2^{64}). La sortie de cette fonction est additionnée avec z5 et le résultat est le code MAC-I.

La fonction de chiffrement f8, réalisé par l'algorithme SNOW 3G est montré par la figure 3.13. La variable d'initialisation est composé des entrés COUNT-C, PORTEUSE et DIRECTION. L'algorithme SNOW 3G génère le keystream, en blocs de taille 32 bits chacun, égale à la taille du message à chiffrer. Le keystream est utilisé pour chiffrer l'information avec un simple XOR.

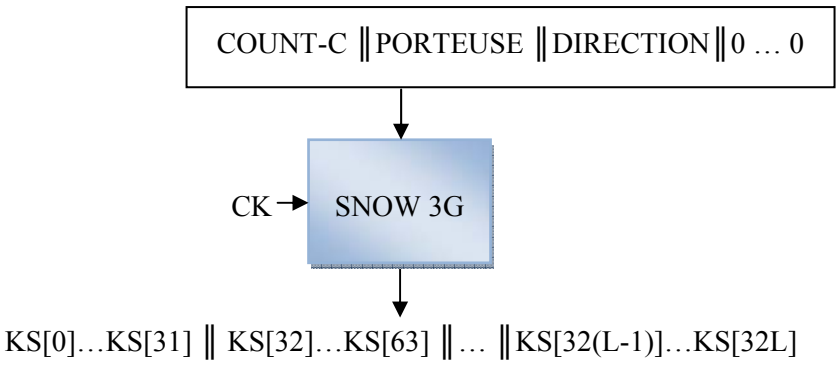


Figure 3.13: Fonction de chiffrement f8 utilisant l'algorithme SNOW 3G (UEA2)

Une attaque contre l'algorithme SNOW 3G a été présentée en 2009 par B. Debraize et I. M. Corbella. Il s'agit d'une attaque qui injecte des erreurs dans le LFSR. L'injection de 22 erreurs permet la récupération de la clé de chiffrement. Des contre-mesures on été proposé pour rendre l'algorithme robuste contre ce type d'attaque. Ces mesures sont prises en compte par l'UMTS.

3.6. Amélioration de la sécurité UMTS

Malgré les faiblesses citées plus haut, qui sont dues à l'interopérabilité avec le standard GSM, à l'utilisation de l'algorithme de KASUMI (algorithme cassé) et à l'envoie en claire de l'identité de l'utilisateur IMSI, la sécurité de l'UMTS est toujours considérée plutôt robuste.

Dans ce paragraphe, nous analysons certains aspects de la sécurité UMTS qui peuvent être des points faibles et nous allons présenter quelques propositions permettant de renforcer la sécurité de l'UMTS.

3.6.1. Transmission sécurisée de l'IMSI

3.6.1.1. Problématique

Un des premiers points faibles identifié de la sécurité UMTS est l'envoie de l'identité permanente IMSI en claire lors de la première connexion RRC (lorsque le mobile est allumé) ou lors d'une panne du VLR. Ceci permet à un attaquant de connaître l'identité de l'abonné et donc d'ouvrir la voie à différentes attaques.

Biais (2006) avait considéré que l'identité de l'abonné n'est pas un aspect important de la sécurité. Cependant, Khan (2008) a montré que l'utilisation du mobile pour effectuer des payements, (commerce électronique) ou pour des transactions bancaires peut amener un attaquant à identifier l'utilisateur et même à déterminer sa localisation. En effet, il existe sur le marché des dispositifs capable d'intercepter l'IMSI des utilisateurs (Khan, 2008) mais leur coût reste très élevé, environ 500.000 dollars, donc, ses dispositifs sont utilisables quand l'information à intercepter vaille le coût.

L'obtention de l'IMSI est très simple: l'attaquant utilise une fausse station de base. Quand l'utilisateur allume son portable, il se connecte à cette station de base qui va lui demander de s'authentifier avec l'identité permanente IMSI. Ce scénario est montré par la figure 3.14 :

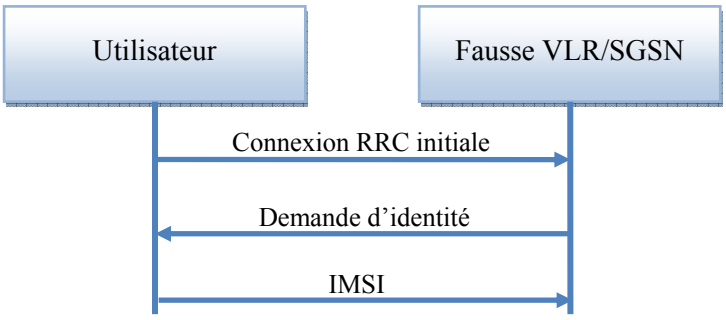


Figure 3.14: Obtention de l'IMSI

3.6.1.2. Solution existante à ce problème

Une solution à ce problème a été proposée en 2006 par Al-Saraireh. Cette solution implique le chiffrement d'IMSI s'il n'y a pas un TMSI qui peut être utilisé. Le chiffrement est réalisé avec une fonction de sécurité spéciale, f10 (3GPP TS 33.103). Nous montrons les étapes de cette procédure dans la figure 3.15.

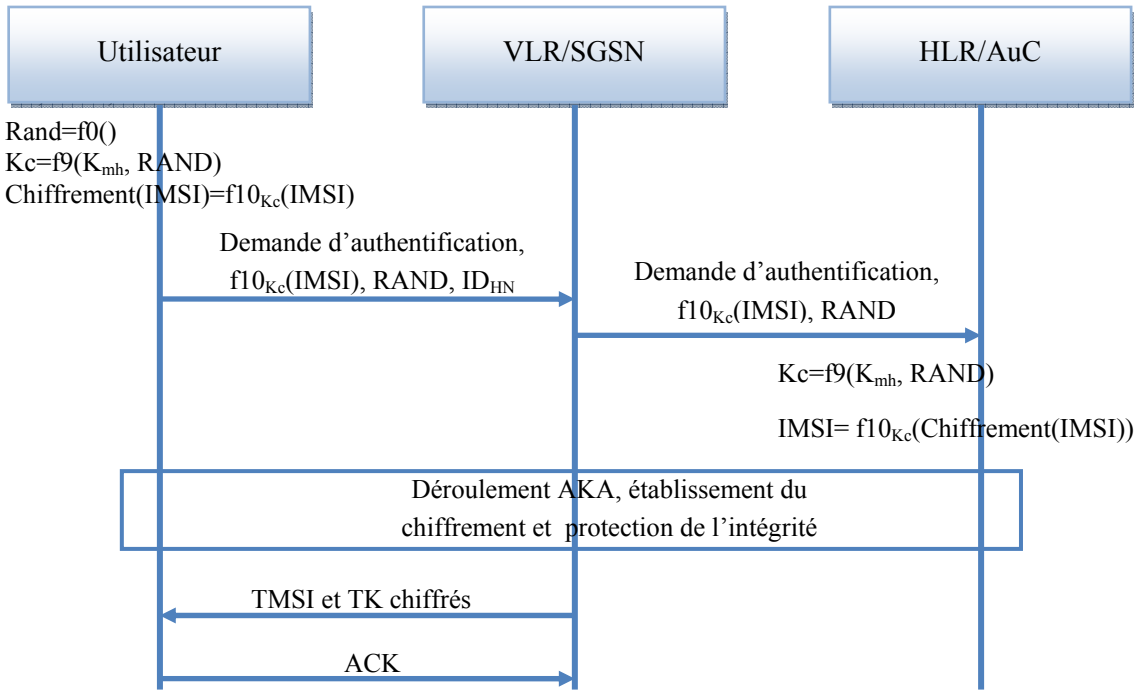


Figure 3.15: Solution proposée par Al-Saraireh pour la sécurité d'IMSI

Quand le mobile est allumé, il ne dispose pas des clés de chiffrement CK et d'intégrité IK. Le standard de sécurité UMTS permet l'envoie des messages sans aucune protection. Néanmoins, même si le pair (CK, IK) n'est pas établi, le mobile dispose d'une clé secrète K, et d'un dispositif de confiance l'USIM qui peut effectuer l'opération de chiffrement en toute sécurité.

La solution d'Al Saraireh suppose que l'USIM du mobile utilise la fonction f0 pour générer une valeur aléatoire RAND. Ensuite la fonction de hachage f9 est utilisée pour la génération d'une clé Kc utilisée pour le chiffrement d'IMSI. La fonction f9 a deux arguments: une clé secrète, K_{mh}, et la valeur RAND. La clé K_{mh} est une clé partagée entre le HLR et tous ses clients. Le HLR utilise la valeur RAND et la clé K_{mh} pour traiter la valeur de la clé Kc. Cette valeur est ensuite utilisée pour le déchiffrement de l'IMSI. Le message envoyé par le mobile au réseau de service (VLR/SGSN) contient l'identifiant de son réseau d'origine, ID_{HN}. Ceci permet au réseau de service de diriger le message vers le bon réseau d'origine. A son tour, le réseau d'origine utilise la valeur reçue RAND et la clé secrète K pour générer la clé Kc, qui sera utilisé pour déchiffrer l'IMSI.

3.6.1.3. Améliorations proposées: Protection de l'intégrité des messages de contrôle

Nous avons vu que la sécurité UMTS protège l'intégrité des messages de contrôle. Nous proposons de respecter cette philosophie pour les messages de contrôle utilisés au cours de l'authentification avec l'IMSI chiffré. La clé Kc peut être utilisée pour vérifier l'intégrité des messages. Pour cela un code d'intégrité, MAC sera calculé pour la clé Kc et le message envoyé au HLR, ($f10_{Kc}$(IMSI), RAND). Ce code sera ajouté au message. La fonction d'intégrité utilisée, f12, sera

choisie de manière qu'elle soit plus rapide que l'algorithme de chiffrement et doit permettre une identification plus efficace de l'abonné, comme nous allons le montrer par la suite.

3.6.1.4. Améliorations proposées: Renoncer à l'utilisation de la clé K_{mh}

La clé secrète K_{mh} est très importante pour la procédure d'authentification utilisant l'IMSI chiffré. C'est une clé partagé entre le HLR et tous ses clients, ce qui permet au HLR de déchiffrer les messages de tous les utilisateurs. Dans le même temps, ceci est une faiblesse parce que si un attaquant trouve la valeur de cette clé (en attaquant un USIM: par cryptanalyse des algorithmes utilisés, par attaque du protocole, par attaque contre le HLR, etc.), alors, il peut casser la confidentialité de l'IMSI pour tous les utilisateurs du HLR en cause. Pour cela nous proposons un protocole qui n'utilise pas une clé partagé par tous les utilisateurs, mais qui demande plus des ressources à l'HLR.

Notre proposition est de remplacer la clé K_{mh} par la clé K, quand la clé Kc est calculée par l'utilisateur. Dans ce cas, pour le HLR, l'identification consiste à trouver la bonne clé K qui a été utilisée par le mobile. Le HLR connaît les mobiles qui sont éteints et va essayer toutes les paires (Ki, IMSIi) qui appartiennent à des mobiles éteints. Si l'IMSI trouvé est l'IMSIi lors de l'utilisation de la clé Ki, alors le mobile est identifié. Sinon, une autre paire (Kj, IMSIj) est essayée. La procédure d'identification du mobile est représentée par l'organigramme de la figure 3.16.

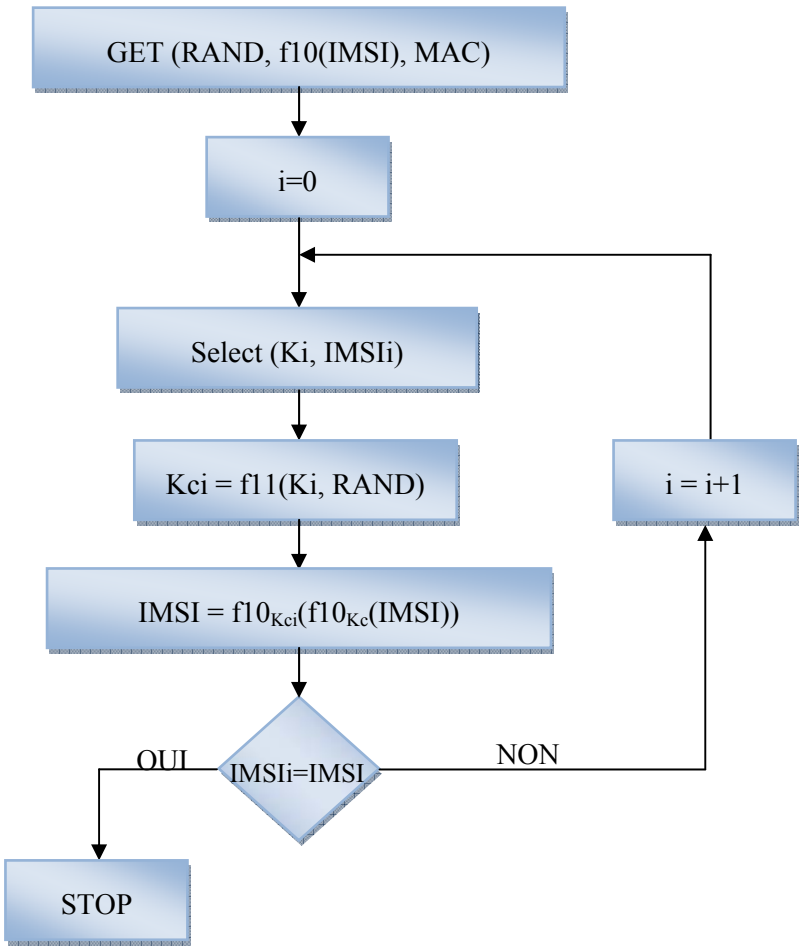


Figure 3.16: Procédure d'authentification du mobile

3.6.1.5. Améliorations proposées: Taille de la clé Kc

La fonction f9, utilisée pour la génération de la clé Kc, génère une sortie sur 64 bits. Une clé de 64 bits utilisée pour chiffrer des données n'offre pas une vraie sécurité. Pour cela nous proposons l'utilisation d'une nouvelle fonction de hachage, f11, pour le traitement de la clé Kc. Pour les réseaux qui utilisent UAI1 ou UAI2, la nouvelle fonction f11, peut utiliser le même algorithme de chiffrement KASUMI ou SNOW 3G mais en plus elle doit être capable de générer une sortie sur 128 bits.

3.6.1.6. Améliorations proposées: Conclusions

Nous proposons dans la figure 3.17 ci-dessous, une variante améliorée de la procédure d'identification d'Al Saraireh.

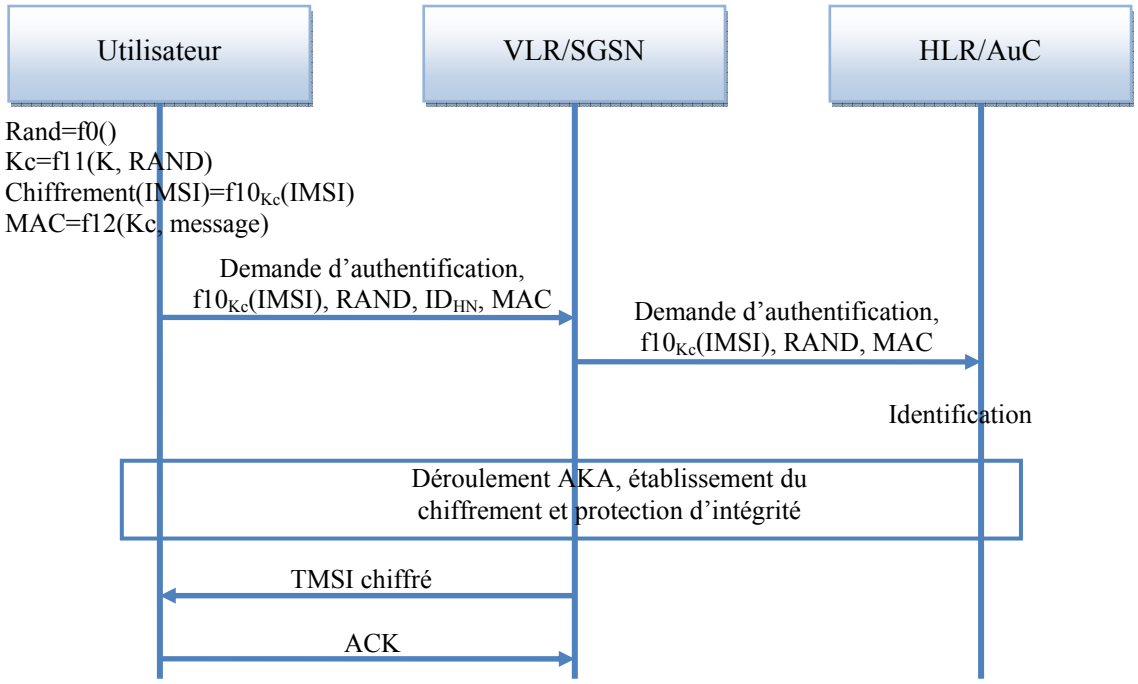


Figure 3.17: Solution améliorée pour la sécurité d'IMSI

Pour prendre en compte la protection de l'intégrité des messages, le HLR va modifier la procédure décrite dans la figure 3.16. Il va ajouter une fonction de comparaison entre le code MACi, calculé avec la clé Ki, et le code MAC reçu. La procédure résultante est décrite dans la figure 3.18.

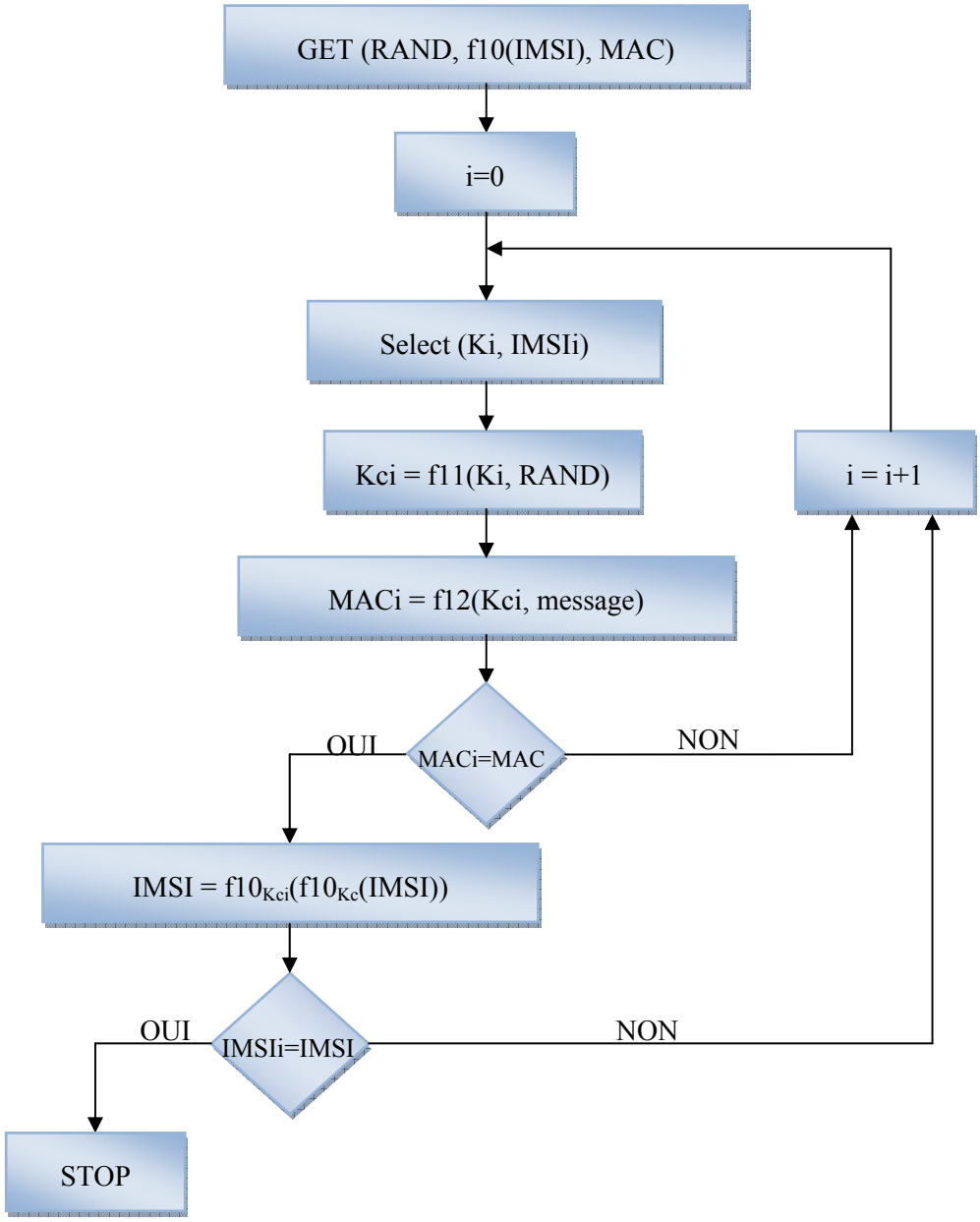


Figure 3.18: Procédure d'authentification pour la solution améliorée

L'avantage de procéder ainsi, découle du fait que la fonction f12, qui calcule le MAC, est plus rapide que la fonction de chiffrement f10. Ceci permettra une identification plus rapide. Si ce n'est pas le cas, la procédure décrite dans la figure 3.16 peut être utilisée. Elle sera suivie par une étape de vérification du code MAC.

Un des problèmes qui reste à étudier est le nombre d'utilisateurs qu'un HLR peut gérer dans ces nouvelles conditions. Le HLR ne sait pas quelle est la valeur de la clé K parce qu'il ne connaît pas l'identité de l'abonné (la procédure qui se déroule est l'authentification de l'abonné). Dans des conditions normales, le nombre d'abonnés qui ont le portable éteint n'est pas très important et le HLR va trouver assez vite la valeur correcte de la clé K.

Par ailleurs, si le HLR tombe en panne et doit se réinitialiser, alors l'authentification de tous les abonnés d'un HLR dans une courte durée peut se révéler comme un défi.

3.6.2. Protection de la clé secrète K

3.6.2.1. Considérations générales

Les attaques les plus dangereuses visent à casser la clé K. Ceci a pour conséquence de compromettre toutes les communications (suivantes et précédentes si le trafic a été enregistré). Un type d'attaque contre la clé K est la cryptanalyse des fonctions de sécurité f1 à f5, utilisées dans la procédure AKA. Les messages qui sont transmis au cours de la procédure AKA peuvent être interceptés par un attaquant qui peut les utiliser pour mener différents types de cryptanalyse.

Nous avons mentionné dans le paragraphe 3.2.2.4, que les attaques de cryptanalyse peuvent être classées en fonction du type de texte qui se trouve à la disposition du cryptanalyste, à savoir: attaque à texte chiffré seulement (ciphertext-only attack), attaque à texte en clair connu (known-plaintext attack), attaque à texte en clair choisi (chosen plaintext attack), attaque à texte en clair choisi adaptatif (adaptive-chosen plaintext attack). En ce qui suit, nous allons voir quels types d'attaques peuvent être montés contre chaque fonction de sécurité. Ceci nous permettra de mettre en évidence les failles de la sécurité UMTS, liées au fait que les fonctions de sécurité sont exposées à des attaques cryptographiques. Cette analyse est présentée dans les paragraphes 3.6.2.2 et 3.6.2.3, et nous proposons, dans les paragraphes 3.6.2.5 et 3.6.2.6, deux méthodes permettant de contrecarrer ces faiblesses.

Au cours de l'exécution de la procédure AKA, les messages sont transmis en clair, ils n'ont aucune protection. En effet, même si la procédure AKA est utilisée pour le changement des clés CK et IK, celles-ci sont effacées avant le début de la procédure, et donc elles ne sont pas utilisées pour protéger les messages échangés au cours de la procédure AKA. C'est vrai que le jeton d'authentification, AUTN, contient un code MAC qui protège son intégrité, mais son importance est très limitée par rapport aux attaques que nous allons décrire dans ce paragraphe et, par suite, son influence peut être même négligée.

Nous allons étudier deux scénarios d'attaque: le premier scénario suppose que l'attaquant intercepte les messages qui sont envoyés sur la voie radio. Le second scénario implique que l'attaquant dispose de la carte USIM et peut intercepter les messages qui sont transmis/reçus par la carte à puce au cours de la procédure AKA.

3.6.2.2. Scénario no. 1 (Vulnérabilité no. 1): Attaques sur la voie radio

Dans ce cas, l'attaquant écoute la voie radio et intercepte les messages entre l'UE et le réseau, au cours du déroulement de la procédure AKA. Ensuite, ces messages sont utilisés pour monter des attaques de cryptanalyse contre les fonctions de sécurité f1 à f5, dans le but de trouver la clé K.

Comme le montre la figure 3.19, l'attaquant dispose des données suivantes : RAND, AUTN= ($SQN_{HE} \oplus AK \| AMF \| MAC$), RES.

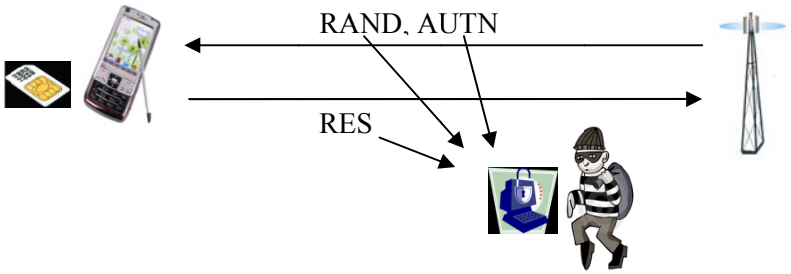


Figure 3.19: Scénario d'attaque sur la voie radio

La figure 3.20 montre comment les données interceptées par l'attaquant sont utilisées dans les fonctions de sécurité f1 à f5.

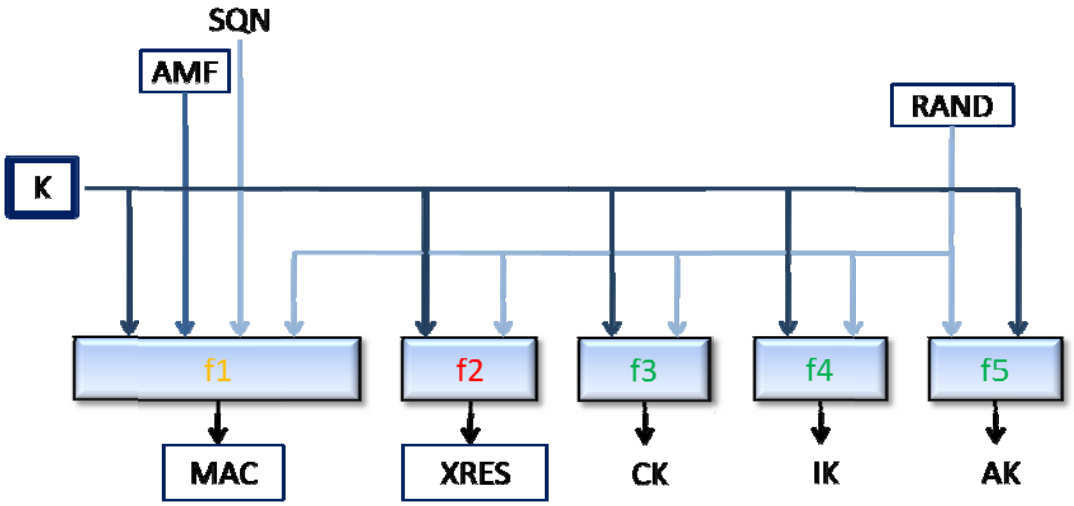


Figure 3.20: Possibilités d'attaque sur la voie radio contre les fonctions de sécurité f1 à f5

L'attaque vise à trouver la valeur de la clé secrète K. Les valeurs des paramètres RAND, MAC, AMF et XRES sont encadrés par un carré pour montrer qu'elles peuvent être interceptées par un attaquant qui écoute la voie radio. En effet, ses valeurs sont envoyées sur la voie radio soit par le mobile, soit par le réseau UTRAN.

Si l'attaquant peut intercepter ces données, il peut alors, monter une attaque à texte en clair connu contre la fonction f2, puisqu'il possède son entrée, RAND, et sa sortie, XRES. Aussi il peut monter une attaque à texte chiffré seulement contre la fonction f1, car il possède sa sortie. Cependant il ne peut mener aucune attaque cryptographique contre les fonctions f3, f4 et f5, parce que l'attaquant ne dispose pas des sorties de ces fonctions. Elles ne sont pas envoyées sûr la voie radio. Pour illustrer nos propos, et par souci de clarté, nous avons colorié, dans la figure 3.20, 'f1' en jaune, 'f2' en rouge (indiquant une faiblesse importante) et les autres fonctions en vert.

3.6.2.3. Scénario no. 2 (Vulnérabilité no. 2): Attaques contre la carte à puce

Un autre type d'attaque pour trouver la clé K qui est mémorisée dans la carte à puce, est d'utiliser un ME modifié permettant à l'attaquant de voir les messages qui sont reçus et envoyés par la carte USIM. L'attaquant utilise ces informations pour monter une attaque de cryptanalyse contre la clé K utilisée lors de la procédure AKA. Une fois la clé K est trouver, l'attaquant peut alors déchiffrer toutes les communications effectuées avec cette carte USIM.

Le scénario de l'attaque contre la carte à puce est montré dans la figure 3.21.

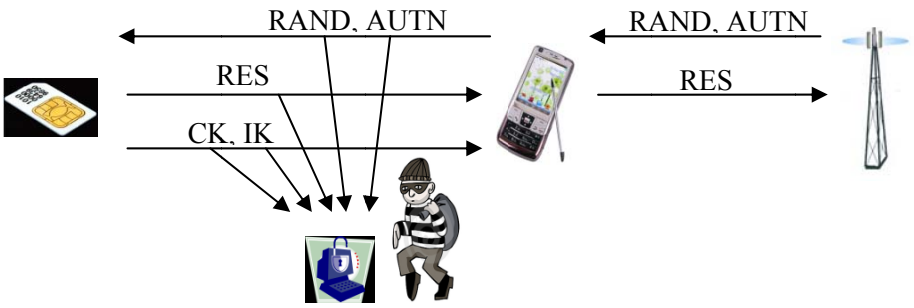


Figure 3.21: Scénario d'attaque contre la carte à puce USIM.

Grâce à l'ME modifié, l'attaquant peut envoyer des AV choisis à la carte à puce pour traitement par celle-ci. Comme la première étape du traitement de l'AV est la vérification du MAC, c'est-à-dire de l'intégrité du message, alors l'attaquant ne peut pas envoyer n'importe quel AV, il doit envoyer seulement des AV valides. Donc, l'architecture de la sécurité UMTS est assez robuste pour empêcher les attaques à texte en clair choisi. En effet, même si l'attaquant peut choisir les entrées (RAND et AUTN) des fonctions de sécurité, il n'aura pas les sorties (RES, CK, IK) correspondantes, parce que l'AV sera ignoré par l'USIM. Par conséquent, les attaques à texte en clair choisi ne sont pas possibles dans ce scénario. La différence notable entre l'attaque sur la voie radio et l'attaque contre la carte à puce, vient du fait que la carte à puce transmet les clés de chiffrement CK et d'intégrité IK au ME. Alors, dans ce cas, l'attaquant peut monter des attaques à texte en clair connu contre les sorties de fonctions de sécurité f3 et f4.

La figure 3.22, montre comment les données interceptées par l'attaquant peuvent être utilisées pour des attaques cryptographiques, si l'attaquant a accès à la carte à puce. Dans cette figure, nous avons encadré les données qui sont à la portée de l'attaquant. Aussi, nous avons colorié: 'f1' en jaune parce qu'elle est exposée à des attaques à texte chiffré seulement, 'f2', 'f3' et 'f4' en rouge parce qu'elles sont exposées à des attaques à texte en clair connu et 'f5' en vert parce qu'elle n'est pas exposée à des attaques cryptographiques.

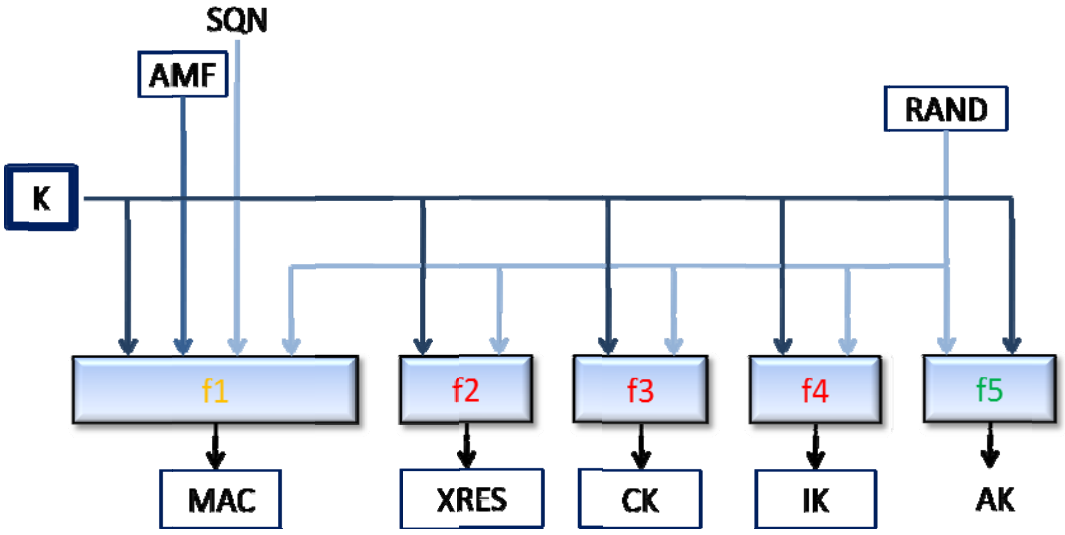


Figure 3.22: Possibilités d'attaque contre les fonctions de sécurité, via les attaques contre la carte à puce

3.6.2.4. Résumé des vulnérabilités

Etant donné l'importance de la clé secrète K, tout attaquant utilise tous les moyennes à sa disposition pour essayer de casser cette clé. Le seul moment, où la clé K est utilisée et les données obtenues sont disponibles à l'attaquant, est au cours de la procédure AKA.

Nous avons étudié les types d'attaques qui peuvent être menés contre chaque fonction de sécurité pour savoir si elle est exposée et avec quel degré ou non. Le tableau 3.2, résume ses différentes attaques contre les différentes fonctions de sécurité au cours de la procédure AKA.

Tableau 3.2 : Types d'attaque auxquels les fonctions de sécurité sont exposées

	Attaque sur la voie radio	Attaque contre la carte à puce USIM
Attaque à texte en clair connu	f2	f2, f3, f4
Attaque à texte chiffré seulement	f1, f2	f1, f2, f3, f4
Attaque à texte en clair choisi	-	-
Attaque à texte en clair choisi adaptatif	-	-

Nous observons, que la fonction f2 est la plus exposée aux attaques. Les fonctions f1, f3 et f4 sont aussi prédisposées à des attaques. C'est seulement la fonction f5, qui semble protégée contre les attaques cryptographiques.

Le standard UMTS et les premières fonctions de sécurité UIA1 et UEA1 ont été proposés en 1999. Durant 11 ans, UIA1 et UEA1 ont été considérés, an tant d'algorithmes de chiffrement et

d'intégrité, très robustes. Cependant, comme nous l'avons déjà mentionné dans le paragraphe 3.4.2.1, une méthode de cryptanalyse rapide a été développée par Dunkelman et al. 2010 permettant de retrouver la clé secrète. Si une telle attaque est réalisée, elle mettra alors, en danger total la sécurité des communications. Comme solution préventive à cette attaque, nous proposons les deux remèdes suivants.

3.6.2.5. Remède no. 1 : Protection des messages d'authentification

Nous avons montré que les fonctions de sécurité utilisées au cours de la procédure d'authentification peuvent être soumises à différents types d'attaque cryptographique. Une manière très économique de se protéger contre ces attaques, est le chiffrement des valeurs RAND, AUTN= ($SQN_{HE} \oplus AK \| AMF \| MAC$), RES envoyées sur la voie radio. Ces valeurs sont traitées par l'application USIM de l'UICC et par le réseau d'origine de l'utilisateur, et elles peuvent être interceptées sur la voie radio quand elles sont envoyées entre le réseau de service et l'équipement mobile. Rappelons, que si une des valeurs $START_{PS}$ ou $START_{CS}$ dépasse la valeur seuil THRESHOLD, le ME effacera les clés CK et IK et déclenchera une nouvelle procédure AKA (en mettant la valeur START de l'USIM à la valeur THRESHOLD et l'indice des clés utilisées, KSI, à la valeur '111').

Notre proposition est simple et très pratique: quand les valeurs $START_{PS}$ ou $START_{CS}$ atteignent ou dépassent la valeur THRESHOLD, le ME déclenchera une nouvelle procédure AKA avant d'effacer la clé CK. Cette clé sera utilisée une dernière fois au cours de la nouvelle procédure AKA pour chiffrer les valeurs RAND, AUTN= ($SQN_{HE} \oplus AK \| AMF \| MAC$), RES, envoyées sur la voie radio.

Cette procédure est très simple à implémenter car elle n'implique qu'un changement dans l'ordre des étapes qui mènent au changement des clés. Le chiffrement des valeurs envoyées sera réalisé par le ME et le RNC avec le même algorithme et la même clé que le chiffrement des données. Donc, la solution que nous proposons n'implique pas l'implémentation d'un nouveau protocole mais seulement une amélioration du protocole déjà mis en place. Ceci permettra une meilleure protection des fonctions de sécurité f1 à f5 car l'attaquant doit commencer son attaque avec la fonction f8, qui est utilisée pour le chiffrement.

3.6.2.6. Remède no. 2 : Renforcement de la clé K

La sécurité UMTS utilise des algorithmes de chiffrement, d'intégrité et d'établissement des clés qui travaillent avec des clés de 128 bits. Conformément aux recommandations en vigueur, une clé de taille 128 bits, assure une protection pour une longue durée, 30 années selon les recommandations ECRYPT II (Smart, 2009).

Cependant, selon l'utilisation des clés et le type des données chiffrées, nous pouvons définir plusieurs niveaux de sécurité requise: le chiffrement des données avec la clé CK assure la

confidentialité des données transportées sur la voie radio. Sa compromission implique la compromission des données en question tant la clé reste inchangée. Donc, on peut dire que la compromission de la clé CK (ou IK) implique une compromission limitée des données supposées confidentielles. Comme ses clés sont changées de temps en temps (dans certains cas pour chaque nouveau appel), alors, ni les données qui ont été envoyées avant l'utilisation de ces clés, ni celles envoyées après ne sont pas compromises.

Ce n'est pas le cas si la clé sécrète K est compromise. En effet, la connaissance de cette clé par un intrus, implique la compromission de toutes les données qui ont été envoyées et qui seront envoyées sur la voie radio. Notons, que les clés CK et K ont la même taille et sont exposées de manière similaire sur la voie radio.

- On envoie beaucoup des données protégées avec les clés CK et IK, mais pas de texte en clair (avec la clé CK).
- On envoie moins de données protégées avec la clé secrète K, mais on envoie de texte en clair aussi, (comme nous l'avons montré dans les paragraphes 3.5.2.2. et 3.5.2.3.).

Nous proposons un changement de l'utilisation de la clé secrète K, inspiré de la procédure d'identification: le standard UMTS limite l'utilisation de l'identité permanente, l'IMSI, le plus possible. Cette identité est utilisée seulement dans des cas précis: pour la première connexion RRC (c'est-à-dire quand on allume le MS) ou lors d'une panne du VLR. Sinon, dans des conditions normales, une identité temporaire le TMSI est utilisée. Nous proposons l'utilisation d'une valeur temporaire de la clé K, TK (Temporary K) pour la procédure AKA.

En plus, pour chiffrer la clé TK avant l'établissement d'une connexion sécurisée, nous proposons d'adapter la solution de chiffrement de l'IMSI, proposée par Al-Saraireh, 2006. A cet effet, l'analyse de la solution d'Al-Saraireh que nous avons déjà effectuée dans le paragraphe 3.6.1.2, nous permet de proposer la procédure améliorée suivante (voir figure 3.23).

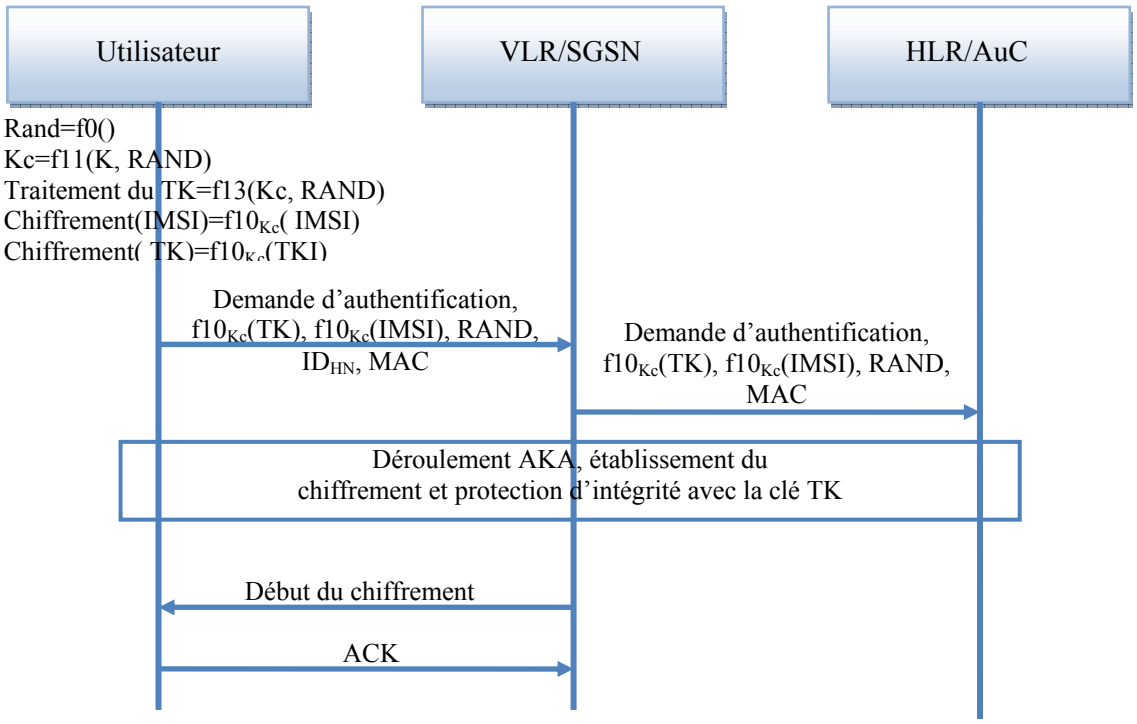


Figure 3.23: Etablissement de la clé TK avec identification par l'IMSI

La fonction f13 est une nouvelle fonction de sécurité, qui reste à être définie. C'est une fonction de hachage et doit être implémentée sur la carte USIM de l'utilisateur et dans le réseau d'origine.

Quand l'utilisateur allume son mobile, ou quand le réseau le lui demande, le mobile doit s'identifier auprès du réseau avec son identité permanente. Pour cela, il doit traiter la valeur RAND et la clé Kc, comme décrit dans le paragraphe 3.6.1.2. En plus, il doit calculer aussi la valeur de la clé temporaire TK. Cette valeur sera calculée avec la fonction f13 de génération des clés, à partir de la clé secrète partagée Kc et du paramètre RAND. La clé TK sera envoyée chiffrée avec la clé Kc. Donc, la procédure proposée est toujours basée sur le secret partagé K, mais limite l'exposition de K car les données transmises sur la voie radio ne sont pas chiffrées avec la clé K en question. En plus on envoie seulement la sortie de la fonction de sécurité f10.

Si l'attaquant surveille la voie radio et intercepte les messages envoyés au cours de la procédure décrite dans la figure 3.23, il va intercepter:

$$f10_{Kc}(TK) = f10_{Kc}(f13(Kc, RAND)) = f10_{Kc}(f13(f11(K,RAND), RAND)) \quad (3.2)$$

Nous pouvons affirmer qu'une cryptanalyse de ce message pour retrouver la clé secrète K n'est pas praticable vue la complexité du calcul.

Toutes les étapes de la procédure AKA restent inchangées. La seule modification faite, par notre proposition d'amélioration de la sécurité, est le remplacement de la clé K par la clé TK. Nous rappelons que, au cours de la procédure AKA, un attaquant peut intercepter l'entré (RAND) et la sortie (RES) de la fonction f2 (voir paragraphe 3.5.2..4). Ceci lui permettra d'envisager une gamme très large d'attaques contre la fonction f2. Ces attaques vont se focaliser maintenant sur la clé TK, au lieu

de la clé K. Alors, si l'attaque est réussie, les données compromises seront limitées aux données sécurisées durant la période de vie de la clé TK.

En plus, nous proposons l'augmentation de la taille de la clé K de 128 bits à au moins 256 bits minimum. Les recommandations d'EUROCRYPT II (Smart, 2009) et de NIST (Barker, 2007) conseillent, pour une sécurité élevée, une longueur minimale de 128 bits pour les clés secrètes. Cette valeur sera la taille: des clés de chiffrement CK, de la clé d'intégrité IK et de la clé TK. Vue l'importance de la clé K, nous proposons d'utiliser une taille minimale de 256 bits au lieu de 128 bits.

3.6.3. Limitation de la confiance dans le réseau d'accueil.

L'InfoSec peut être défini comme la science qui utilise une « confiance initiale », la propage et l'élargie, pour obtenir une « confiance généralisée ». La confiance initiale est mise dans les algorithmes de sécurité, dans les protocoles de sécurité qui utilisent ces algorithmes et dans les équipements (éléments de réseau) utilisés pour implémenter les protocoles, prises individuellement. La confiance généralisée est donnée par la réunion des éléments de confiance initiale et est composée par les services de sécurité ainsi offerts. Ces services doivent répondre à la politique de sécurité du réseau en question.

S'il n'y a pas de confiance initiale aucun service de sécurité ne peut pas être offert. Jusqu'à maintenant nous avons étudié les algorithmes et les protocoles concernant la confiance initiale pour les réseaux UMTS. En ce qui suit, nous allons étudier les éléments du réseau.

La procédure AKA se déroule entre le mobile, l'USIM plus précisément, et le réseau de service, le VLR ou le SGSN. Elle permet à l'un d'authentifier l'autre. Cela est possible parce que les deux entités USIM et VLR/SGSN ont un point en commun: chacune d'elle a déjà réalisé une authentification mutuelle avec le réseau d'origine, le HLR/AuC.

La communication entre le HLR/AuC et le VLR est réalisée avec des réseaux fixes qui utilisent des protocoles bien connus, tels que l'IP, l'ATM, le Frame Relay, etc. C'est la sécurité de ces protocoles qui permet l'authentification entre le HLR/AuC et le VLR. En ce qui concerne l'authentification entre l'utilisateur et le HLR /AuC, elle se base sur l'application USIM de la carte à puce qui a été fournie à l'utilisateur par le réseau d'origine et qui contient la clé K.

Si le réseau de service appartient au même operateur de téléphonie mobile, ou au même pays que le réseau d'origine, l'utilisateur peut faire confiance totale à la procédure AKA telle qu'elle est. Mais si le réseau de service appartient à un opérateur auquel l'utilisateur ne fait pas confiance ou s'il se trouve dans un autre pays, l'utilisateur est contraint par la procédure AKA d'avoir confiance dans le réseau de service.

Avec le terrorisme accru dans le monde, il y a des nouvelles lois qui autorisent les réseaux de service de certains pays d'intercepter et de décrypter tous les messages transmis au sein de son réseau. Ceci légalise en quelque sorte l'écoute indiscrète de toutes les informations transmises au sein des

réseaux et crée un manque de confiance entre l'utilisateur et les réseaux mobiles, surtout si ces dernières accueillent provisoirement l'utilisateur mobile.

L'interception légale reste un problème délicat, car il est au croisement du droit des individus à une vie privé et des considérations de sécurité. La plupart des pays sont dotés d'une législation dans ce domaine. Nous allons étudier le cas de l'Europe.

La résolution 96/C 329/01 a été la première publiée par la Communauté Européenne sur ce sujet. Elle reconnait le droit des états à intercepter des communications dans le cadre de leur sécurité nationale. La CE a mis en évidence les besoins suivants:

- L'accès à l'ensemble des données transmises depuis ou vers un abonné.
- L'accès à l'ensemble des données reliées à l'appel dont la signalisation, l'identité de l'appelé et de l'appelant, l'information de localisation, l'information sur les services et paramètres spécifiques de l'utilisateur.
- La possibilité de surveillance en temps réel.
- Une ou plusieurs interfaces pour les systèmes de surveillance des services officiels qui délivrent en clair toutes les communications, mêmes celles chiffrées.

L'interception doit avoir les caractéristiques suivantes:

- L'interception doit être transparente pour l'utilisateur.
- La fiabilité de l'interception doit ou être au moins au même niveau que la fiabilité du service qui intercepte.
- La mise en place d'une interception doit pouvoir être réalisée dans des courts délais.
- Les données interceptées doivent être protégées contre toute autre utilisation que celle des autorités en charge.

Nos propositions s'inscrivent dans le cadre de la législation européen et ne touchent pas aux fonctions du réseau d'origine. Cependant, elles vont permettre de donner à l'UE de l'utilisateur et au réseau d'origine, la possibilité d'initier certaines procédures pour être informés des certains aspects de la sécurité de la connexion mais pas plus que ça. Car le réseau de service doit se soumettre aux lois nationales.

3.6.3.1. Choix des algorithmes

Une des procédures où l'utilisateur et le réseau d'origine sont contraints d'avoir confiance dans le réseau de service, est la procédure d'établissement des algorithmes de chiffrement et d'intégrité. Cet aspect devient très important si on sait que les algorithmes UIA1 et UEA1 ont déjà été cassé.

Nous avons discuté en détail le déroulement de l'établissement d'une connexion sécurisé dans le paragraphe 3.3.4.3. Nous résumons dans la figure 3.24, la partie liée au choix des algorithmes de chiffrement et d'intégrité.

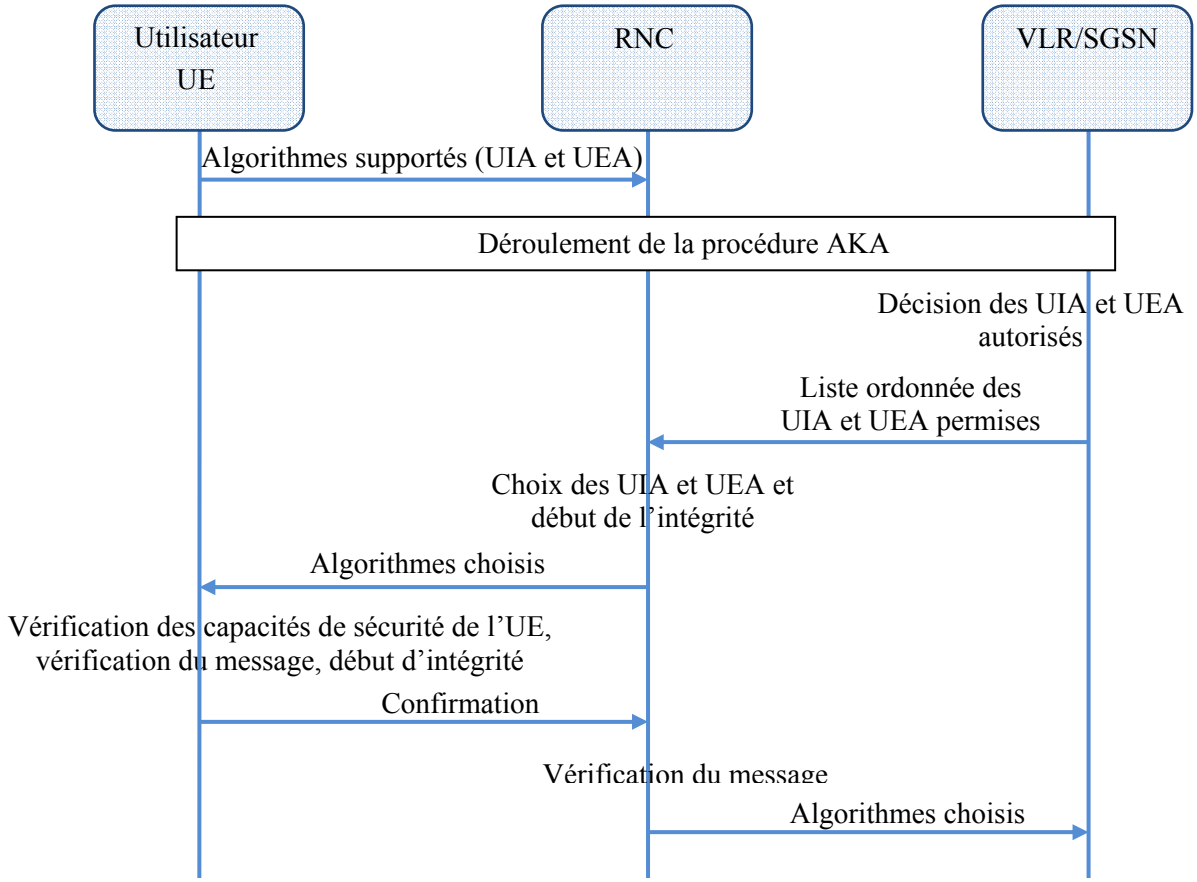


Figure 3.24: Etablissement des algorithmes de chiffrement et d'intégrité

Dans cette procédure on remarque l'absence totale du réseau d'origine. En résumé, le RNC fait le compromis entre les algorithmes supportés par le ME de l'utilisateur et les algorithmes permis par le VLR/SGSN. Ceci permet au VLR de choisir un algorithme de sécurité ou même renoncer au chiffrement, s'il choisi UEA0.

La procédure que nous proposons implique aussi le réseau d'origine et lui donne un rôle important. C'est le réseau d'origine qui est informé des capacités de l'UE de l'abonné, qui décide des algorithmes autorisés et de leur ordre de préférence et qui est aussi informé des algorithmes choisis par le RNC. En plus nous proposons que les messages échangés entre l'utilisateur et le réseau d'origine aient leur intégrité protégée. Ceci est simple à réaliser avec la clé secrète K (TK, si la proposition du paragraphe 3.5.2.6. est utilisée) et la fonction de sécurité f12.

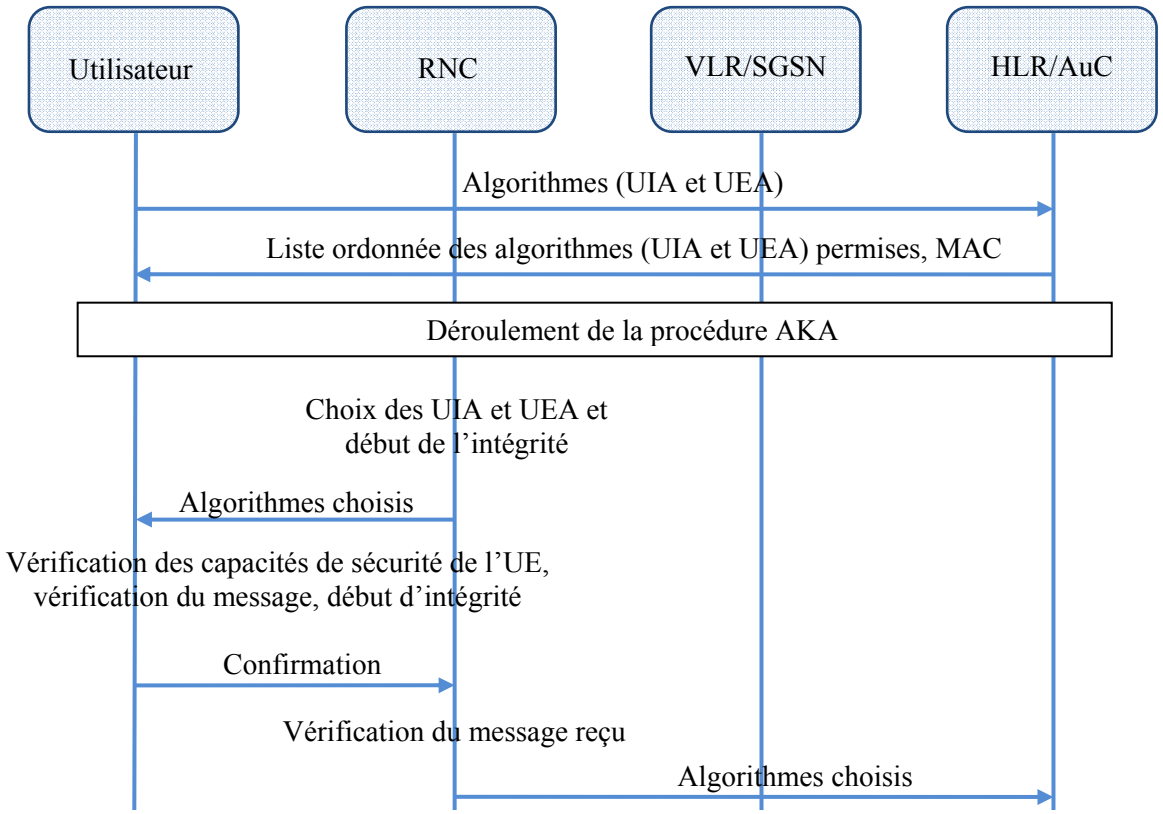


Figure 3.25: Procédure proposée pour l'établissement des algorithmes de chiffrement et d'intégrité

Bien sûr, la communication entre le mobile et le HLR passe par le RNC et le VLR. Ils recevront aussi ses messages. Ceci permet à l'RNC de choisir les algorithmes qu'il supporte, immédiatement après la procédure AKA et au VLR d'être informé sur les algorithmes qui sont utilisés.

Nous pouvons remarquer que c'est toujours le RNC qui fait le choix des algorithmes, mais cette fois la procédure est transparente à l'utilisateur et au réseau d'origine de l'utilisateur. Ceci permet au réseau d'origine d'évaluer le comportement du réseau de service. Cette évaluation peut mener à l'interdiction d'utiliser certains réseaux de service.

3.6.3.2. Changement du TMSI

Le changement du TMSI est réalisé pour protéger l'identité de l'utilisateur. Nous avons discuté l'importance de la protection de l'identité dans le paragraphe 3.5.1. Nous avons vu que si l'identité de l'utilisateur n'est pas bien cachée, alors, l'utilisateur peut être suivi.

Dans le standard UMTS, le changement du TMSI est réalisé seulement à l'initiative du réseau de service (3GPP TS 23.060). Ce dernier envoie à l'UE le nouveau TMSI et le LAI et l'UE confirme la réception. Cette procédure est montrée dans la figure 3.26.

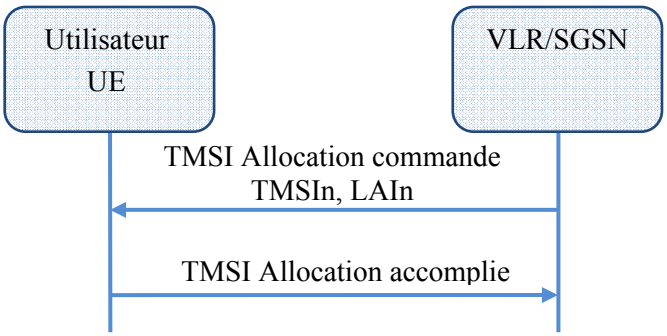


Figure 3.26: Procédure standard du changement du TMSI

Pour renforcer la sécurité du TMSI, nous proposons l'utilisation d'un mécanisme qui, par rapport au standard actuel, donne à l'USIM et au réseau d'origine la possibilité d'initier le changement du TMSI. Le mécanisme en question rapporte chaque changement du TMSI au réseau d'origine. De cette manière, toutes les parties impliquées peuvent suivre ou initier le mécanisme.

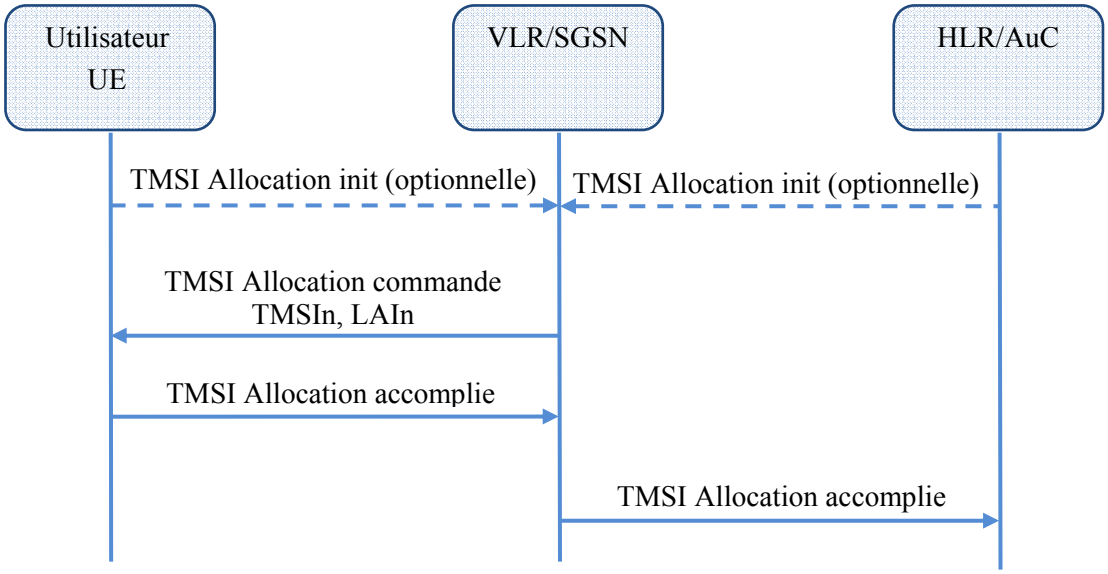


Figure 3.27: Procédure proposée pour le changement du TMSI

Les deux changements que nous avons proposés, font que les actions du réseau de service soient visibles pour le réseau d'origine et permettent à l'UE et au réseau d'origine d'initier le changement du TMSI. Ceci permet d'augmenter la confiance accordée par le réseau d'origine et l'utilisateur, au réseau de service, sans diminuer l'importance de ce dernier.

3.7. Conclusion

Dans ce chapitre nous avons étudié la sécurité des réseaux de communications mobiles de troisième génération l'UMTS. Nous avons présenté la philosophie de la sécurité 3G et nous avons vu qu'elle s'inspire largement de la sécurité 2G. La sécurité 3G conserve les avantages de la sécurité 2G

et traite ses faiblesses, donc, la sécurité 3G peut être vue comme une variante améliorée de la sécurité 2G.

Ensuite nous avons étudié l'architecture de la sécurité pour les réseaux UMTS. Nous avons vu que la sécurité UMTS a cinq composantes : la sécurité au niveau accès, la sécurité du domaine réseau, la sécurité du domaine utilisateur, la sécurité du domaine application, la visibilité et la configuration de la sécurité. Nous avons montré comment ces cinq composantes travaillent ensemble pour offrir la sécurité dans tout le réseau UMTS.

Nous avons étudié en détaille et proposé des améliorations pour trois aspectes de la sécurité UMTS. Le premier aspect abordé de la sécurité a été la protection de l'identité permanente IMSI. Nous avons montré que cette identité peut être compromise avec l'utilisation d'une fausse station de base et nous avons montré la solution proposée par Al Saraireh pour ce problème. Ensuite, nous avons proposé trois améliorations pour cette solution qui la rende plus robuste.

Le deuxième aspect traité de la sécurité a été l'étude de la vulnérabilité de la clé secrète K. Nous avons étudié les scénarios qui permettent une attaque cryptographique contre la clé K et nous avons montré pourquoi nous considérons qu'elle est trop exposée par rapport aux conséquences de sa compromission: la compromission total de la sécurité des données passées et futures. Nous avons proposé deux mesures qui protègent la clé K contre ces attaques: le chiffrement des messages de la procédure AKA et l'utilisation d'une valeur temporaire de la clé, TK, de manière similaire avec l'utilisation de l'identité temporaire TMSI.

Enfin, nous avons abordé la problématique de la confiance dans le réseau de service. Chaque utilisateur doit faire plus de confiance dans le réseau d'origine que dans le réseau de service. Cependant, les procédures de sécurité laissent au réseau de service des taches très importantes comme le choix des algorithmes de chiffrement ou le changement de l'identité temporaire TMSI. Nous avons proposé le changement des protocoles de sécurité pour permettre au réseau d'origine et à l'UE d'être informés sur les choix et les actions du réseau de service. Ceci permet au réseau d'origine et à l'UE d'évaluer le comportement des réseaux de service.

4. Tatouage numérique

4. Tatouage numérique

4.1. Introduction générale

La croissance des réseaux informatiques à grand débit, en général, et l'Internet, en particulier, a permis des échanges de données pour des applications très variées touchant aux domaines des affaires, sciences, sociales et activités récréatives, etc.... La nécessité de protection de l'information numérique est apparue avec le développement des communications à grande échelle des différents standards de télécommunications incluant l'Internet.

Quand il s'agit de l'information numérique, c'est très difficile de faire une distinction claire entre l'original et les copies, ou de dépister les modifications malveillantes subies par les données informatives. La voie est ouverte pour la violation du droit d'auteur ou pour les modifications malicieuses.

Dans ce contexte, la technique de tatouage numérique représente une nouvelle technologie qui peut être utilisée afin de résoudre les disputes de droit d'auteur ou de tester l'intégrité et l'authenticité du contenu numérique des multimédias. Il permet ainsi le remplacement des sceaux, timbres et signatures analogiques par leurs versions numériques.

Le tatouage numérique représente un message porteur d'information inséré dans un média-hôte (une image, un signal audio ou un signal vidéo), et qui offre une modalité de vérification d'intégrité des données ou un mécanisme de revendication des droits d'auteur.

Afin d'être efficace, un tatouage numérique doit être imperceptible, statistiquement invisible et robuste. Le tatouage numérique est dit imperceptible lorsque la qualité visuelle de l'image tatouée n'est pas affectée par rapport à l'image originale. Un tatouage est dit statistiquement invisible lorsque l'extraction du tatouage, à base de plusieurs données numériques tatouées avec la même clé n'est pas possible par des méthodes d'attaques statistiques. Enfin, un tatouage numérique est dit robuste si après traitement, intentionnel ou non, la partie essentielle du tatouage reste dans l'image traitée.

Le tatouage fragile peut être utilisé pour remplacer les sceaux parce qu'il assure le service d'intégrité des données. Il permet aux personnes autorisés de vérifier si l'image qu'ils regardent est l'image originale ou si elle a été modifiée. Par exemple, la figure 4.1. a) montre une image originale et la figure 4.1 b) l'image modifiée. Si l'image originale a été tatouée, une personne autorisée peut savoir, en traitant l'image modifiée, qu'une personne et les bagages ont été effacés par un attaquant.

a) Image originale *b) Image modifiée*

Figure 4.1: Utilisation du tatouage fragile.

Le tatouage robuste est utilisé pour renforcer les droits d'auteur. Il peut être utilisé pour la simple revendication des droits d'auteur, ou pour contrôler la distribution du contenu multimédia. Si le tatouage est utilisé pour la revendication des droits d'auteur, le tatouage inséré est un texte qui indique la personne ou l'institution qui à le droit d'auteur. Si le tatouage est utilisé pour le contrôle de la distribution du contenu, le tatouage indique aussi la personne au quelle le contenu a été distribué. Si cette personne distribue de manière illégale le contenu, l'interception du contenu permet de savoir qui est responsable de sa distribution illégale.

4.2. Généralités sur le tatouage numérique dans le cas des images JPEG

Nous rappelons ci-dessous les différentes étapes du standard JPEG ainsi que l'emplacement de l'insertion/extraction du tatouage sur les coefficients DCT quantifiés. Nous justifions, aussi, les raisons de l'insertion du tatouage, uniquement sur la composante luminance Y de l'image YCbCr.

4.2.1. Présentation simplifiée du standard JPEG

Le standard JPEG peut comprimer n'importe quels formats d'images, mais en pratique le format luminance/chrominance, YCbCr, est le plus utilisé, parce que l'œil humain est plus sensible aux variations de la composante luminance (Y) qu'aux variations de la couleur, représentée par les composantes chrominance bleu Cb, et chrominance rouge Cr. Cb et Cr donnent des informations sur les variations des couleurs. Par rapport au format RGB, où tous les composantes ont la même importance, la composante Y du format YCbCr contient l'information visuelle la plus importante pour l'œil humain. Ceci permet un très fort taux de compression pour les deux autres composants. Afin d'obtenir la composante luminance, l'image est convertie en image couleur RGB, puis l'image RGB est transformée à son tour en image YCbCr.

Les grandes étapes de la compression JPEG sur la composante Y sont :

- Le découpage en blocs : 8x8 pixels (découpage le plus utilisé).
- Transformation DCT : pour séparer les fréquences basses (qui représentent en gros la valeur moyenne de l'image, et donc l'information la plus sensible visuellement) et les fréquences

hautes (qui représentent les détails fins de l'image, et donc l'information la moins importante visuellement).

- Quantification : pour éliminer les hautes fréquences, une matrice de quantification est utilisée. Pour chaque bloc obtenu, les 64 coefficients DCT sont calculés, puis quantifiés en utilisant la formule suivante:

$$AC_{Qkl}=\text{round}\,(AC_{kl}/Q(k,l));\ k,l \in \{1,2,\ldots,8\} \tag{4.1}$$

où AC_{kl} représentent les coefficients DCT non quantifiés, AC_{Qkl} représentent les coefficients DCT quantifiés et [Q] est la matrice de quantification. La matrice de quantification typique, spécifié dans le standard JPEG original est donnée par:

$$[Q]=\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{4.2}$$

- Codage entropique : pour coder l'information utile obtenue après la quantification, le codage par plages suivi par le codage d'Huffman sont utilisés.

Le processus inverse, de décodage, consiste à faire les étapes suivantes :

- Décodage entropique.
- Dé-quantification.
- Transformation DCT inverse.
- Reconstitution de la composante Y de l'image.

4.2.2. Schéma général de la procédure du tatouage numérique sur les coefficients DCT quantifiés.

Pour l'application qui nous intéresse, et en vue des arguments donnés ci-dessus sur l'importance de la composante Y, le tatouage numérique est inséré uniquement à la composante luminance (Y), après l'opération de quantification et avant le codage entropique.

Les différentes étapes du standard JPEG et l'insertion/extraction du tatouage sont montrées dans la figure suivante :

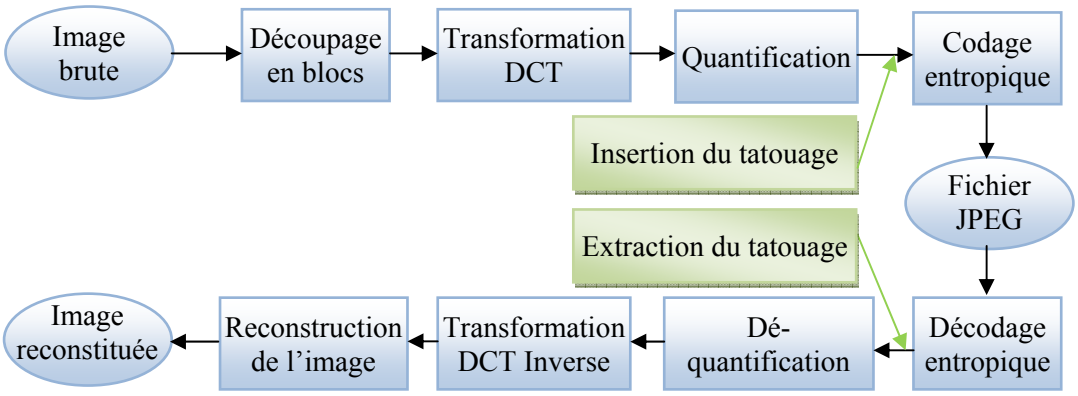


Figure 4.2. Standard JPEG et insertion/extraction du tatouage

4.2.3. Considérations sur l'implémentation du tatouage numérique

Le tatouage numérique peut être inséré soit dans un fichier JPEG source existant, soit dans le plan RGB d'une image qui ensuite sera transformée dans un fichier JPEG.

Dans le cas d'un fichier JPEG source (par exemple une image stockée dans un ordinateur, ou une image transmise via l'Internet), pour avoir le fichier JPG tatoué, on procède comme indiqué sur la figure 4.2. On lit les coefficients DCT quantifiés du fichier JPEG source (c'est-à-dire, en gros, on effectue le décodage entropique) puis on applique la procédure d'insertion du tatouage numérique. Enfin, on écrit les nouveaux coefficients DCT quantifiés et tatoués dans un fichier JPEG (c'est-à-dire, en gros, on effectue le codage entropique).

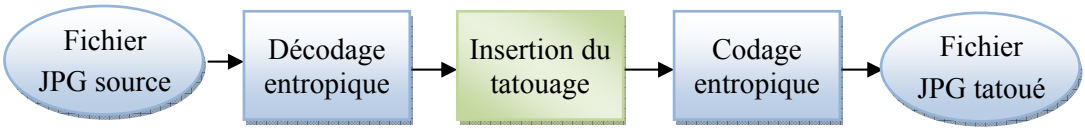


Figure 4.3. Insertion du tatouage à partir d'un fichier JPEG

Dans le cas d'une image RGB (par exemple une image fournie par les capteurs d'une camera numérique) le tatouage est réalisé après l'opération de quantification et avant le codage entropique comme indiqué sur la figure 4.4.

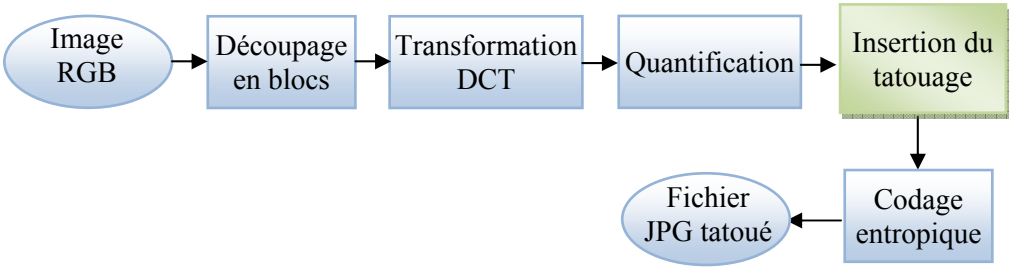


Figure 4.4. Insertion du tatouage à partir d'une image source RGB

La vérification de l'intégrité de l'image est réalisée à partir de l'image tatouée et produit une image qui indique les régions altérées. Bien évidement, le décodage entropique est utilisé pour obtenir les coefficients DCT quantifiés tatoués comme indiqué sur la figure 4.5.

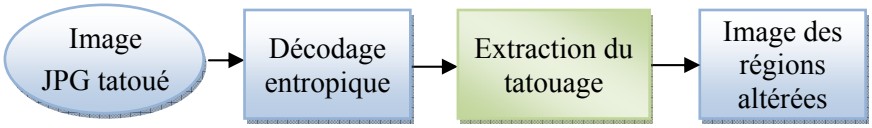


Figure 4.5. Vérification de l'intégrité de l'image tatouée

Dans la suite de ce travail, nous nous intéressons aux méthodes de tatouage numérique fragile basées sur les signaux chaotiques pour la vérification de l'intégrité des données. Plus particulièrement, dans un premier temps, nous analysons en détail le travail de Wang (2008) à propos d'un algorithme de tatouage fragile. L'algorithme en question possède certains avantages manifestes liés à sa structure simple et à son interopérabilité avec le standard JPEG. Cependant, nous montrons qu'il présente des points faibles qui nous ont permis de le cryptanalyser. Ensuite, nous proposons un modèle basé sur la chaîne de Markov permettant de calculer la probabilité de casser l'algorithme de Wang en fonction du nombre d'images tatouées utilisées. Cette analyse nous a permis de développer un nouvel algorithme de tatouage fragile basé chaos qui conserve les avantages de l'algorithme de Wang, et semble robuste contre différentes techniques de la cryptanalyse.

4.3. Description détaillée de l'algorithme de Wang (2008)

Wang, et al. (2008) ont proposé un algorithme de tatouage numérique fragile basé sur le chaos pour l'intégrité des images JPEG. L'algorithme travaille sur les coefficients DCT quantifiés de l'image JPEG.

Le fait que l'algorithme travaille directement sur les coefficients DCT quantifiés lui confère une très bonne interopérabilité avec le standard JPEG et permet une détection du tatouage très efficace en réception.

4.3.1. Processus d'insertion du tatouage numérique

Pour chaque bloc, l'algorithme utilise tous les coefficients AC quantifiés qui ont une valeur plus grande que 1, afin de générer et insérer le tatouage numérique. Le coefficient DC situé dans le coin supérieur gauche du bloc reste inchangé.

La figure 4.6 montre le principe de l'algorithme de tatouage.

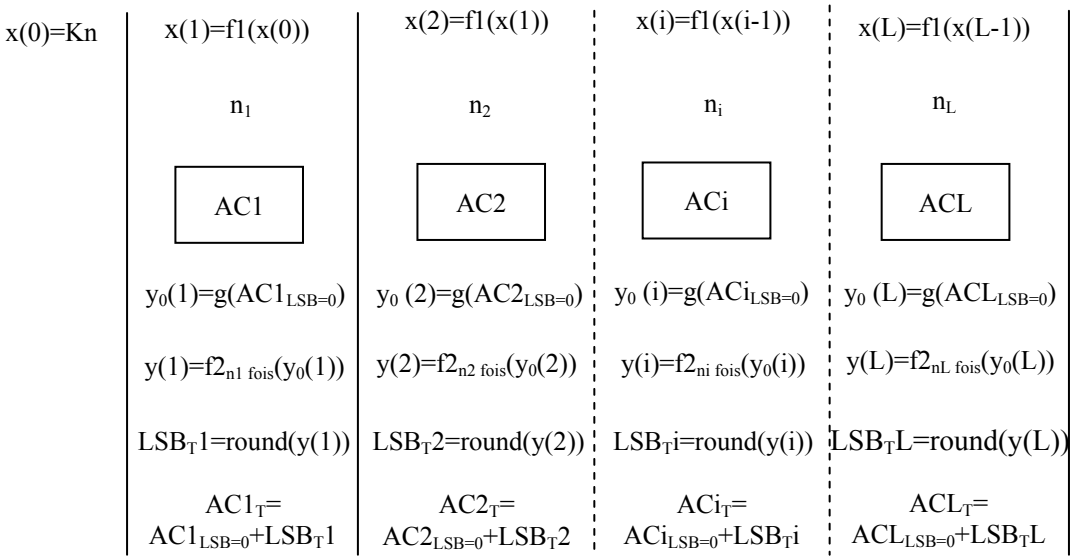


Figure 4.6 : Algorithme de Wang

Dans cette procédure, seulement les coefficients AC supérieurs à 1 sont tatoués ; la justification à cela est donnée en fin de ce paragraphe. La procédure de tatouage se déroule comme suit :

Pour chaque coefficient *ACi*, i=1,…L, le LSB (Least Significant Bit) est mis à zéro, et la valeur ainsi obtenue, $ACi_{LSB=0}$ est utilisée par la fonction *g* dont la sortie $y_0(i)$ sert comme condition initiale d'un système chaotique (système chaotique 2, f2). Ce système chaotique est itéré n_i fois, ($y(i)=f2_{ni\ fois}(y_0(i))$). La valeur n_i est obtenue à partir d'un autre système chaotique (système chaotique 1, f1) qui est itéré une seule fois pour chaque nouveau coefficient. Les sorties de cette fonction chaotique sont calculées comme suit : x(i) = f1(x(i-1)),avec i=1,…L et x(0)= *Kn*, la clé sécrète.

La sortie y(i) du système chaotique 2 est comparée à un seuil S=0,5. Si la sortie est supérieure à S, le bit de tatouage LSB_Ti, est mis à « 1 », autrement il est mis à « 0 ». Le coefficient tatoué ACi_T s'obtient en ajoutant le bit LSB_Ti à la valeur du coefficient $ACi_{LSB=0}$. Le résultat de cette opération constitue l'étape d'insertion du tatouage numérique bloc par bloc.

Les raisons de ne pas tatouer que les coefficients AC supérieurs à 1 sont liées aux taux de compression et à l'ambiguïté qu'on peut avoir lors de la procédure d'extraction. En effet, les forts taux de compression, avec une bonne qualité de l'image du standard JPEG, sont dûs à l'existence d'un nombre important de coefficients AC quantifiés nuls, groupés dans la zone des fréquences supérieures et alors le codage par plages est très efficace. Donc, si nous tatouons ces coefficients, alors, la zone en question contiendra des coefficients de valeurs distribuées aléatoirement entre 0 et 1. Ceci détruira de façon dramatique l'efficacité du codage par plage et par conséquent, le taux de compression JPEG de l'image tatouée sera inférieur au taux de compression JPEG de l'image originale. Pour cette raison, nous évitons de tatouer les coefficients AC nuls.

Par ailleurs, si nous ne tatouons pas les coefficients AC nuls, nous ne pouvons pas tatouer les coefficients AC=1 non plus. En effet, le résultat du tatouage d'un coefficient AC=1 est un coefficient AC_T=0 ou AC_T=1. La procédure d'extraction du tatouage étant aveugle, alors, après la lecture d'un coefficient AC=0 de l'image tatouée, il est impossible de savoir si ce coefficient est porteur d'une information de tatouage (AC_T=0) ou c'est un coefficient AC=0 non concerné par la procédure de tatouage.

Le fichier JPEG est alors créé par le bloc du codage entropique utilisant tous les coefficients DC en claire, et AC tatoués (coefficients strictement plus grands que 1), et AC non-tatoués (coefficients AC inférieurs ou égaux à 1).

Les deux systèmes chaotiques utilisés par l'algorithme de Wang sont basés sur la fonction logistique donnée par:

$$x_{n+1}=\mu \cdot x_n \cdot (1- x_n) \qquad (4.3)$$

où x_n est un nombre réel compris entre 0 et 1 et μ est un nombre positif supérieur à 3.57 et strictement inférieur à 4.

4.3.2. Processus d'extraction du tatouage numérique

Le processus d'extraction du tatouage pour la vérification de l'intégrité de l'image traitée a des points en commun avec le processus d'insertion du tatouage, qui a été détaillé ci-dessus.

La différence principale réside dans le processus d'extraction qui, contrairement au processus d'insertion, mémorise les LSB des coefficients quantifiés tatoués. Ensuite, une comparaison est faite entre les LSB des coefficients tatoués quantifiés et les LSB_T des coefficients quantifiés obtenus par les étapes de génération du tatouage. Pour chaque bloc de 64 coefficients, dès qu'un LSB est trouvé différent de son LSB_T correspondant, le bloc entier est considéré comme falsifié et sera coloré en blanc. Si tous les LSB des coefficients quantifiés tatoués d'un bloc sont identiques à leurs LSB_T correspondants, le bloc en question est considéré comme intègre et il sera coloré en noir. Ceci nous permet d'obtenir directement les régions altérées de l'image.

Le principe de l'extraction est montré dans la figure 4.7.

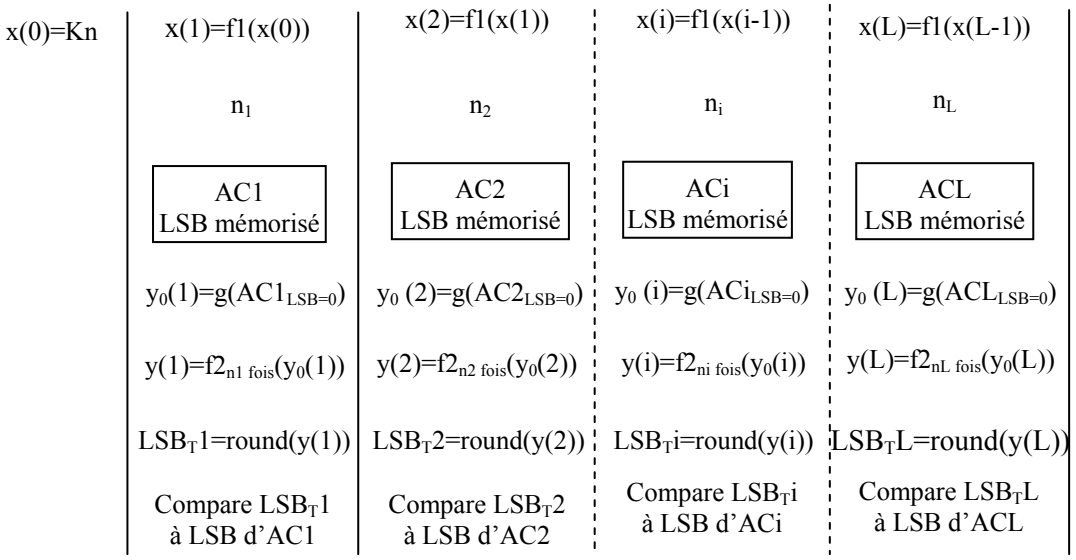


Figure 4.7 : Principe de l'extraction du tatouage numérique

4.4. Cryptanalyse de l'algorithme de Wang (Caragata et al., 2010)

Dans le cas du tatouage fragile, la cryptanalyse essaye de rendre inefficace le processus de vérification de l'intégrité de l'image tatouée. L'attaque la plus dangereuse consiste à passer une image falsifiée comme une image non altérée après le processus de vérification de l'intégrité. Cette attaque se déroule comme suit (voir figure 4.8) : l'attaquant dispose de l'image tatouée et il la modifie à sa guise ensuite, il la tatoue à nouveau. Ceci suppose qu'il a à sa disposition l'algorithme du tatouage ou un algorithme équivalent. Le résultat est une image tatouée attaquée, qui semble intègre lors de l'extraction du tatouage.

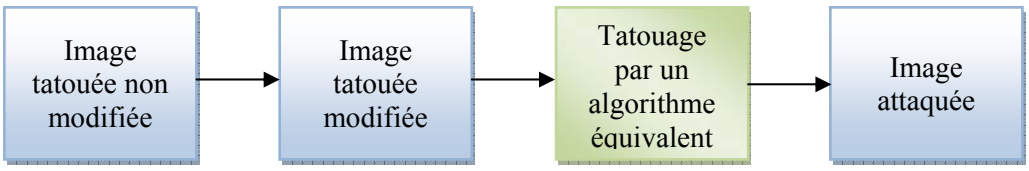


Figure 4.8. : Attaque du tatouage fragile.

Nous avons déjà vu que l'algorithme de Wang présente des avantages intéressants puisqu'il traite les coefficients DCT quantifiés et utilise des fonctions chaotiques qui lui assurent plus de robustesse. Cependant, l'algorithme de Wang présente quelques inconvénients néfastes. Un de ces inconvénients est qu'il n'utilise pas des fonctions chaotiques robustes. G. Alvarez (2006) a étudié l'importance des bonnes propriétés cryptographiques des fonctions chaotiques pour la sécurité. Pour cela et à ce propos, nous proposons au paragraphe 4.4.1 d'utiliser des fonctions chaotiques nettement plus robustes que la fonction Logistique utilisée par l'algorithme de Wang. Un autre inconvénient est

l'ajout de l'information de tatouage sur tous les coefficients AC. Au paragraphe 4.4.2, nous discutons les désavantages de ce choix et l'importance d'ajouter l'information de tatouage sur les coefficients AC qui ont une valeur plus grande qu'une valeur seuil et aussi sur le coefficient DC.

Enfin, l'inconvénient fatal de l'algorithme de Wang est sa faiblesse contre l'attaque que nous développons ci-dessous.

4.4.1. Description de la cryptanalyse proposée

Le premier système chaotique fournit sur sa sortie, après une seule itération, le nombre n_i d'itérations à faire par le deuxième système chaotique.

Les valeurs, n_i, $i=1,2,...,L$ représentent les sorties du système chaotique 1, obtenues en fonction de la clé secrète *Kn*. Ainsi, si le système utilise la même clé, les mêmes valeurs n_i, $i=1,2,...,L$ seront générées pour le tatouage des images différentes. Par ailleurs, la valeur maximale n_{max} de n_i, $i=1,2,...,L$ est de l'ordre de quelques dizaines au plus. Wang et al, proposent l'implémentation de leur algorithme dans des caméras digitales. Nous avons implémenté l'algorithme avec Matlab 7.0.1 sur un ordinateur doté d'un processeur Intel Dual Core avec 2 GB mémoire de RAM. Le temps requis pour tatouer une image de 1160x1600 pixels est supérieur à une minute pour n_{max} égal à 23.

L'attaquant tente de trouver les différentes valeurs n_i, $i=1,2,...,L$ utilisant un certain nombre d'images tatouées. Avec chaque image, il essaye les valeurs plausibles pour chaque élément n_i, comme décrit par la suite. Cette attaque semble être similaire à une attaque exhaustive. La différence est qu'au lieu d'essayer les différentes valeurs de la clé *Kn*, l'attaquant essaye les valeurs des éléments n_i.

Nous procédons la cryptanalyse comme suit:

1) Pour chaque coefficient AC plus grand que 1 de la première image tatouée utilisée, l'attaquant mémorise le LSB_T et la valeur du coefficient, dont le LSB est mis à zéro ($ACi_{LSB=0}$), qui sert à calculer la condition initiale du système chaotique 2.

2) Pour chaque n_i, l'attaquant essaye toutes les valeurs plausibles et calcule la sortie du système chaotique 2 correspondante à chacune de ces valeurs. Chaque sortie (comprise dans l'intervalle [0, 1[, est comparée à un seuil fixé à la valeur 0.5. Le résultat de cette comparaison génère un bit de valeur égale à 1, si la sortie en question est supérieure ou égale à la valeur du seuil. Autrement, la valeur du bit est égale à zéro (cette opération est strictement identique à l'opération du calcul du bit de tatouage donnée par la figure 4.6). La valeur de ce bit est comparée à la valeur du bit LSB_T qui a été mémorisé au début de la procédure. Si les valeurs de deux bits en question sont différentes, alors, la valeur du n_i courante est considérée non plausible et elle est éliminée, sinon elle est conservée.

3) L'attaquant répète les étapes 1) et 2) sur un certain nombre d'images tatouées. Ceci lui permettra pour chaque nouvelle image testée de trouver, pour chaque coefficient ACi, les valeurs n_i non plausibles et de les éliminer. Cette procédure est arrêtée lorsqu'il ne reste qu'une seule valeur plausible par coefficient. Pour plus de clarté, nous illustrons dans la figure 4.9 le déroulement de la cryptanalyse pour un seul coefficient AC*i*.

4) L'attaquant dispose ainsi de la liste de valeurs n_i lui permettant de tatouer des images falsifiées, et alors, il sera impossible de prouver que l'image n'est pas intègre malgré sa falsification. Notons enfin que la cryptanalyse a pu être faite sans connaissance de la clé secrète.

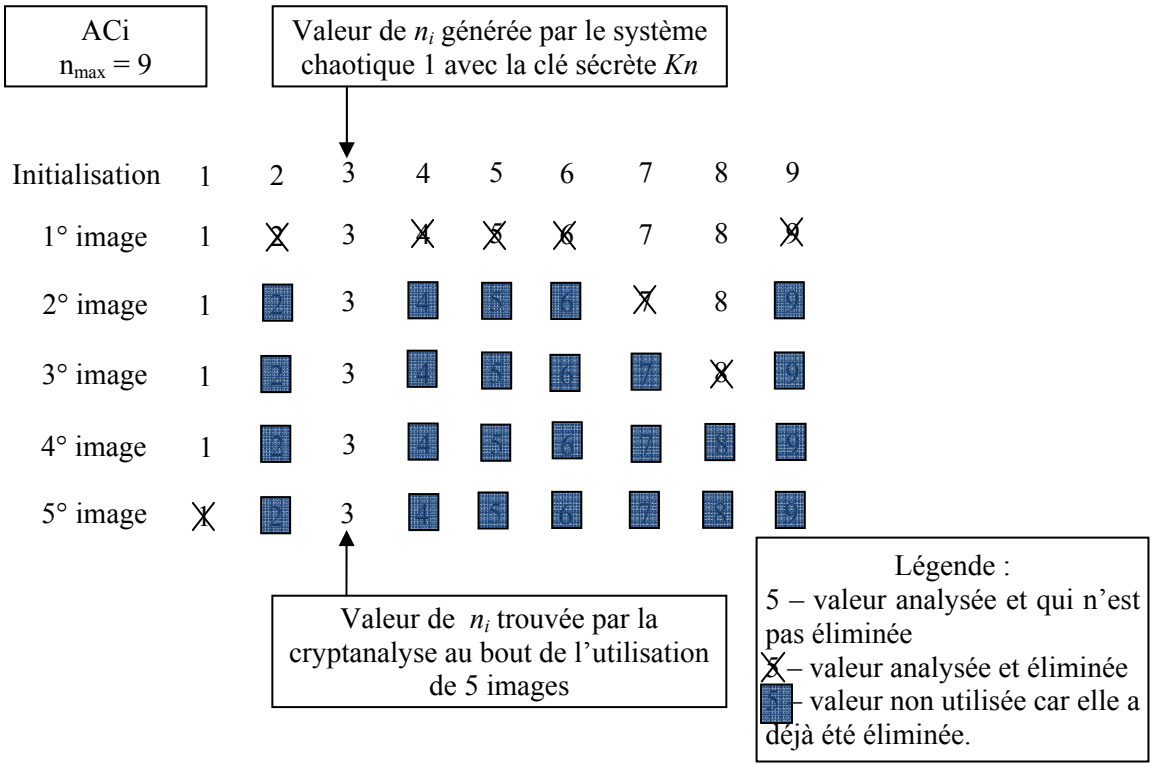


Figure 4.9 : Déroulement de la cryptanalyse pour un seul coefficient

La figure 4.9 permet d'observer que chaque image utilisée au cours de la cryptanalyse, a pour effet l'élimination d'environ la moitié des valeurs plausibles du n_i. En effet, chaque valeur n_i plausible, sauf la valeur cherchée, a une probabilité de 0.5 d'être éliminée. Donc, le nombre d'images à utiliser pour n'avoir qu'une seule valeur plausible pour n_i est approximativement égal à $\log_2(n_{max}-1)$.

Par ailleurs, nous pouvons supposer que la probabilité de trouver la valeur n_i utilisée par l'algorithme de tatouage, en fonction du nombre d'images utilisées, contient trois zones distinctes (voir figure 4.10):

- **Zone de probabilité très proche de la valeur zéro** : Zone qui correspond à un nombre d'images utilisées nettement plus petit que $\log_2(n_{max}-1)$. Par exemple, la probabilité de trouver n_i après avoir utilisé une seule image est exactement égale à $2^{-(nmax-1)}$.
- **Zone de transition** : Zone qui correspond à un nombre d'images utilisées telle que la probabilité de trouver la valeur de n_i ne peut pas être approximée par des valeurs zéro ou un.
- **Zone de probabilité très proche de la valeur un** : Zone qui correspond à un nombre d'images utilisées, telle que la probabilité de trouver n_i peut être approximée par la valeur un.

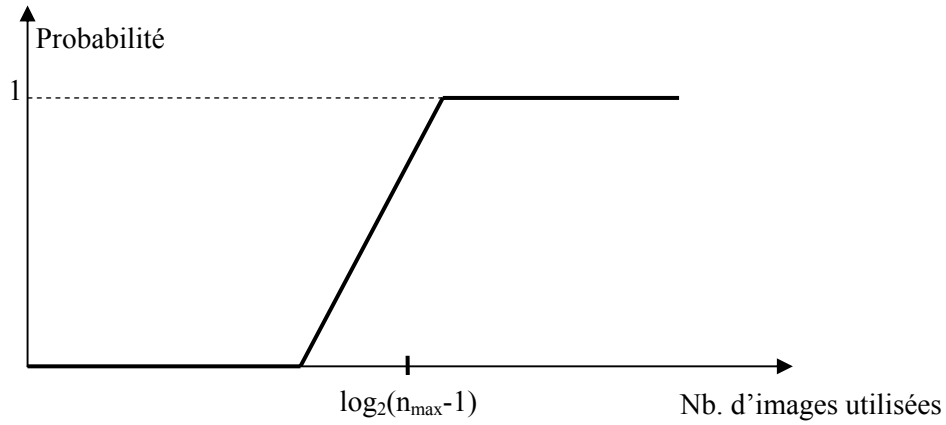


Figure 4.10 : Probabilité de trouver n_i en fonction du nombre d'images utilisées.

Ci-dessus, nous avons analysé l'effet de l'attaque pour un seul coefficient AC. Mais en pratique l'attaque essaye de trouver les valeurs n_i correspondantes à tous les coefficients AC d'une zone d'image tatouée ou de toute l'image tatouée.

Le tableau 4.1 montre un scénario possible de l'effet de l'attaque pour plusieurs coefficients AC :

Tableau 4.1 : Scénario possible de l'attaque sur 8 coefficients AC

	AC1	AC2	AC3	AC4	AC5	AC6	AC7	AC8
Initialisation	9	9	9	9	9	9	9	9
1° image	4	3	5	4	6	4	7	4
2° image	3	1	3	3	3	2	4	1
3° image	2		1	3	1	1	3	
4° image	2			1			2	
5° image	1						2	
6° image							1	

L'objectif de ce scénario est de montrer d'une part, que le nombre d'images nécessaires pour trouver les valeurs n_i (i=1,…8) permettant de casser l'algorithme de tatouage est variable d'un coefficient à un autre et d'autre part, que le graphe (voir figure 4.11) de la probabilité de casser l'algorithme de tatouage en fonction du nombre d'images a une forme similaire au graphe de la figure 4.10, sauf que la « zone de transition » est déplacée vers la droite. Cette constatation (validée plus loin par des simulations) est valable pour une zone donnée de l'image ou pour toute l'image sous traitement.

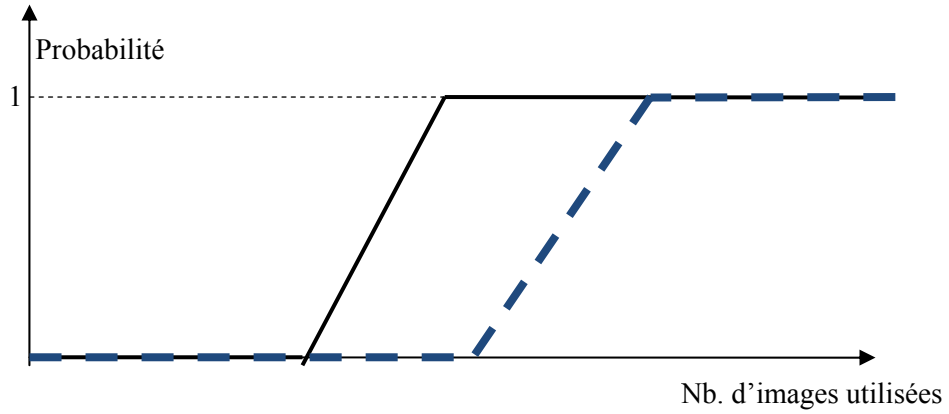


Figure 4.11 : Probabilité de trouver les valeurs n_i (i=1,…L) en fonction du nombre d'images utilisées, pour une zone donnée de l'image ou pour l'image entière.

Dans la suite, nous proposons d'utiliser la chaîne de Markov pour modéliser la cryptanalyse. Cela va nous permettre, à la fois, de confirmer les observations ci-dessus et de tracer avec précision le graphique de la figure 4.11.

4.4.2. Chaîne de Markov (Bremaud, 2008)

Un processus de Markov est un processus dont la probabilité d'apparition d'une situation dépend des situations déjà passées, ainsi que de leur instant. Dans la suite, nous nous intéressons à la chaîne de Markov d'ordre 1, caractérisée par la relation suivante :

$$P\{Z_m = s_{k_m}/Z_1 = s_{k_1}, Z_2 = s_{k_2}, \cdots, Z_{m-1} = s_{k_{m-1}}\} = P\{Z_m = s_{k_m}/Z_{m-1} = s_{k_{m-1}}\} \qquad (4.4)$$

La chaîne de Markov est utilisée pour designer un processus de Markov à temps discret et avec un ensemble fini de situations. Elle est utilisée dans de nombreux domaines tels que : les télécommunications, la génétique, les réseaux, etc….

Soit un processus stochastique pouvant prendre au moment *m* une situation donnée parmi un ensemble de *K* situations possibles :$s_{j_m} \in \{S\} = \{s_{1_m}, s_{2_m}, \cdots, s_{K_m}\}$. Soit *T={1,2,3,…,M}* le domaine du temps, à chaque instant $m \in T$, on définit un vecteur de probabilité de situations formé de *K* éléments $X_m = \{x_{1_m}, x_{2_m}, \cdots, x_{K_m}\}$. Chaque élément x_{i_m}, représente la probabilité que le système soit dans la situation s_i au moment *m*, avec $\sum_{i=1}^{K} x_{i_m} = 1$ et $x_{i_m \geq 0}$.

Par ailleurs, soit P_{ij} , la probabilité de transition du processus de la situation S_i à la situation S_j, avec *i, j = 1,…,K*. Les différentes probabilités de transition forment la matrice de transition du système, donnée par:

$$[P] = \begin{pmatrix} P_{11} & P_{12} & \ldots\ldots & P_{1K} \\ P_{21} & P_{22} & \ldots\ldots & P_{2K} \\ \vdots & & P_{ij} & \\ P_{K1} & P_{K2} & \ldots\ldots & P_{KK} \end{pmatrix} \qquad (4.5)$$

Le vecteur de probabilité des situations à l'instant *m* est donné par la relation suivante:

$$X_m = X_{m-1} \cdot [P] = X_0 \cdot [P]^m \text{ (si } X_m \text{ est stationnaire)} \qquad (4.6)$$

Cette relation décrit la propriété fondamentale de la chaîne de Markov. Elle permet ainsi de déterminer le vecteur de probabilité des situations à l'instant *m* à partir seulement, du vecteur de probabilité initial (à l'instant zéro, X_0) et de la matrice de transition.

4.4.3. Modélisation de la cryptanalyse de l'algorithme de Wang par la chaîne de Markov d'ordre 1

Dans le cas de l'algorithme de Wang, le processus de la cryptanalyse consiste à trouver les valeurs du vecteur $\{n\} = \{n_1, n_2, \ldots, n_l, \ldots, n_L\}$ qui sont générées par l'algorithme de tatouage au cours de son exécution. La cryptanalyse utilise le fait que, si la clé n'est pas changée, l'algorithme va générer des vecteurs *{n}* identiques quel que soit l'image traitée. Cela permet de monter une attaque exhaustive sur le vecteur *{n}* au lieu de la clé secrète.

Quand la cryptanalyse commence, l'attaquant sait seulement que n_l a une valeur entre 1 et n_{max}. C'est-à-dire, que n_l a n_{max} valeurs plausibles. Puis, durant l'attaque, le nombre des valeurs plausibles pour n_l diminue avec chaque image tatouée utilisée. Ce processus d'élimination continue jusqu'à l'obtention d'une seule valeur plausible, donc la valeur cherchée.

Nous utilisons la chaîne de Markov pour modéliser l'attaque sur un seul coefficient, AC_l. Plus précisément, nous calculons la probabilité de trouver la valeur de n_l en fonction du nombre d'images utilisées. Ensuite, ce résultat est généralisé au cas d'une zone de l'image ou pour l'image entière (contenant L coefficients) sous l'hypothèse que les corrélations entre les différentes valeurs n_l ($l=1, 2,...$) sont supposées nulles. Cette hypothèse permet de calculer la probabilité de casser l'algorithme pour une zone d'image contenant L coefficients à partir de la probabilité de casser un seul coefficient. Cependant, ceci est le cas le plus défavorable en ce qui concerne la probabilité de casser le système en fonction du nombre d'images. En effet, une connaissance sur les corrélations, permettra d'accélérer la procédure de casser le système.

En ce qui suit, nous allons expliquer comment appliquer les notions théoriques de la chaîne de Markov sur notre démarche de cryptanalyse. Ici, la chaîne de Markov est un processus à temps discret. Chaque moment de temps correspond au nombre d'images utilisées par le processus de cryptanalyse. En effet, le passage d'un moment donné au moment suivant correspond à l'utilisation d'une image tatouée par l'attaquant. C'est-à-dire, partant du moment initial zéro, le moment m correspond à l'utilisation de m images.

Chaque situation S_j, prise dans l'ensemble des situations $\{S\}= \{S_1, S_2,..., S_K\}$ du processus représente j valeurs plausibles pour n_l. Alors le nombre total des situations du système est $K=n_{max}$ et le processus se trouve dans la situation $s_{n_{max}}$ au moment initial. L'attaque s'arrête quand le processus est dans la situation S_1.

L'élément x_{i_m} du vecteur de probabilité des situations $X_m = \{x_{1_m}, x_{2_m}, \cdots, x_{K_m}\}$ représente la probabilité d'avoir i valeurs plausibles pour n_l après avoir utilisé m images pour la cryptanalyse. Cela veut dire que le vecteur de probabilités des situations initial est :

$$X_0 = (0, 0, 0, ..., 1) \tag{4.7}$$

Nous devons calculer la matrice [P] des transitions du système.

Le processus ne peut qu'éliminer une partie des valeurs plausibles. Il ne peut pas ajouter de nouvelles valeurs plausibles. Alors, $P_{ij} = 0$ si $i<j$. Pour calculer P_{ij} quand $i \geq j$ nous utilisons la définition d'une probabilité dans le cas d'une expérience, dont les résultats sont équiprobables : le rapport entre le nombre des cas favorables Cf et le nombre des cas possibles Cp.

$$P_{ij}=Cf/Cp \tag{4.8}$$

Quand le système passe du moment m au moment $m+1$, l'attaque consiste à analyser l'ensemble des valeurs plausibles du n_l au moment m, afin d'éliminer une partie de cet ensemble. Si le processus

est dans la situation S_i au moment m l'attaque va analyser les i valeurs plausibles du n_l. Pour la valeur cherchée du n_l, (la valeur vraie), la probabilité qu'elle soit éliminée est zéro. Pour les autres valeurs, cette probabilité est égale à 0.5. Alors, le nombre des cas possibles de transition au moment $m+1$, Cp, est donné par:

$$Cp=2^{i-1} \tag{4.9}$$

Par ailleurs, le nombre des cas favorables, c'est-à-dire les cas pour lesquels le processus passe dans la situation S_j au moment $m+1$, Cf, est donné par :

$$Cf = C_{i-1}^{j-1} = C_{i-1}^{i-j} \tag{4.10}$$

Ceci représente le nombre des cas qui conservent j valeurs dont une est la valeur cherchée ou qui éliminent $i-j$ valeurs plausibles pour n_l. Remplaçant maintenant le numérateur et le dénominateur de la relation (4.8), par les expressions données respectivement par les relations (4.9) et (4.10), nous obtenons :

$$P_{ij}=\begin{cases} \dfrac{C_{i-1}^{j-1}}{2^{i-1}}, & i \geq j \\ 0, & i < j. \end{cases} \tag{4.11}$$

Soit, (avec $C_n^p = \dfrac{n!}{p!(n-p)!}$) :

$$P_{ij} = \begin{pmatrix} 1 & 0 & 0 \ldots\ldots 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \ldots\ldots 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \ldots\ldots 0 \\ \vdots & \vdots & \vdots \\ \frac{1}{2^{n_{max}-1}} & \frac{C^1_{n_{max}-1}}{2^{n_{max}-1}} & \frac{C^2_{n_{max}-1}}{2^{n_{max}-1}} \ldots\ldots \frac{1}{2^{n_{max}-1}} \end{pmatrix} \tag{4.12}$$

Nous connaissons déjà le vecteur initial de probabilité des situations (équation (4.7)) et la matrice de transition (équations (4.11) ou (4.12)). Utilisant la relation (4.6) nous pouvons calculer facilement le vecteur de probabilités de situations pour tous les moments qui nous intéressent.

Le premier élément du vecteur de probabilités de situations représente la probabilité qu'il y a une seule valeur plausible pour n_l, c'est-à-dire que l'attaque est finie, et qu'elle a été menée avec succès.

La figure 4.12 montre l'organigramme qui décrit la méthode de calcul de la probabilité de réussite de casser un coefficient et un ensemble de coefficients en fonction du nombre d'images utilisées.

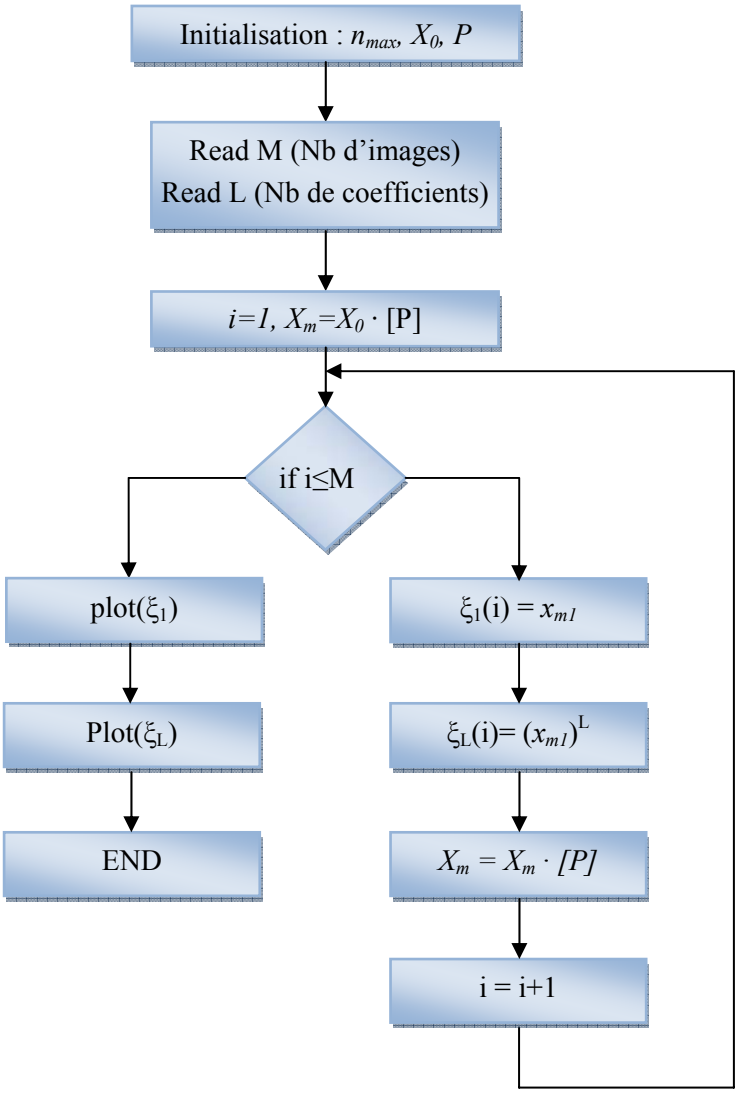


Fig.4.12. Organigramme du calcul de la probabilité de réussite de casser le système

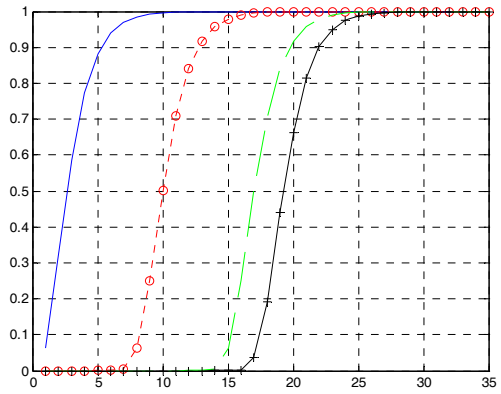
Dans cette figure, $\xi_1(i)$, *et* $\xi_L(i)$ $i=1..M$ représentent respectivement les vecteurs qui contiennent les probabilités de réussite de casser un coefficient AC donné et de casser une zone contenant L coefficients AC tatoués, en fonction du nombre d'images tatouées.

4.4.4. Résultats de la modélisation par les chaînes de Markov

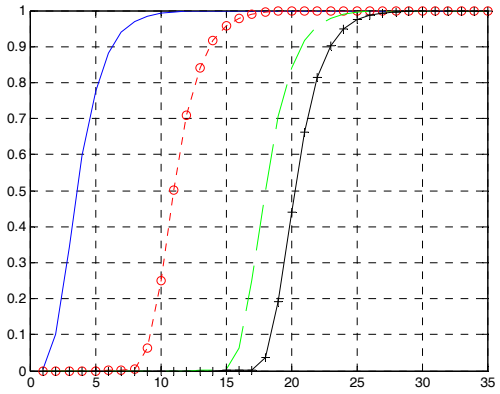
Nous avons utilisé le mécanisme détaillé plus haut pour calculer la probabilité de casser l'algorithme en fonction du nombre d'images utilisées, pour quatre valeurs du paramètre n_{max} : 5, 9, 17, et 33. Nous avons choisi ces valeurs car elles représentent les valeurs plausibles de n_{max} en fonction de l'application (temps réel ou non) et de l'implémentation réalisée (caméra, ordinateur,…). Nous avons déjà vu au paragraphe 4.3.1., que l'algorithme de tatouage devient lent dès que n_{max}=23, ceci nous permet de fixer la limite supérieure de n_{max} à la valeur 33 (valeur justifiée par les résultats de simulation obtenus ci-dessous). Une valeur de n_{max} inférieure à 5, met la sécurité en danger.

Dans les figures 4.13 et 4.14, nous présentons les résultats obtenus. L'axe des abscisses indique le nombre d'images utilisées par l'attaquant et l'axe des ordonnées donne les probabilités (ξ_1, ξ_L) de réussite de la cryptanalyse. Afin d'interpréter plus finement les courbes obtenues, nous avons utilisé pour l'axe des probabilités, une échelle linéaire dans la figure 4.13, et une échelle logarithmique dans la figure 4.14.

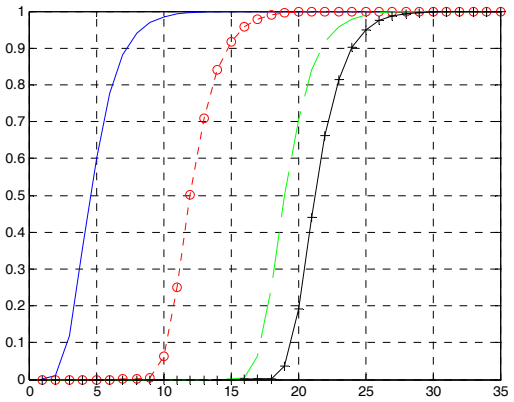
Dans toutes les figures, la courbe en trait continu (couleur bleue) représente le cas de l'attaque sur un seul coefficient, la courbe en cercles (couleur rouge) représente la probabilité de casser l'algorithme pour un bloc de taille 70×70 pixels de l'image tatouée (177 coefficients AC), la courbe en traits pointillés (couleur verte), représente la probabilité de casser l'algorithme pour l'image de la figure 4.18 qui a une taille de 720×600 pixels (22767 coefficients tatoués) et la courbe avec des «+» (couleur noire) représente le cas de l'image de la figure 4.19, qui a une taille de 1160x1600 pixels, soit 108008 coefficients tatoués.



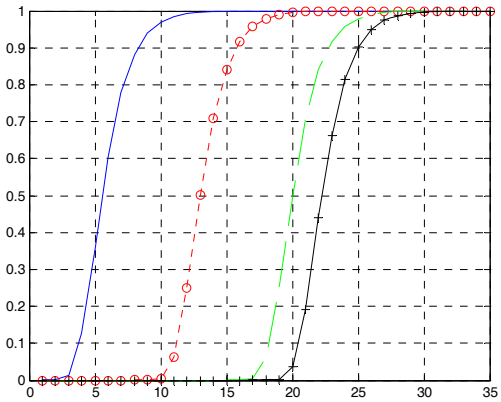
a) n_{max}*=5*



b) n_{max} *=9,*



c) n_{max} *=17*



d) n_{max} *=33*

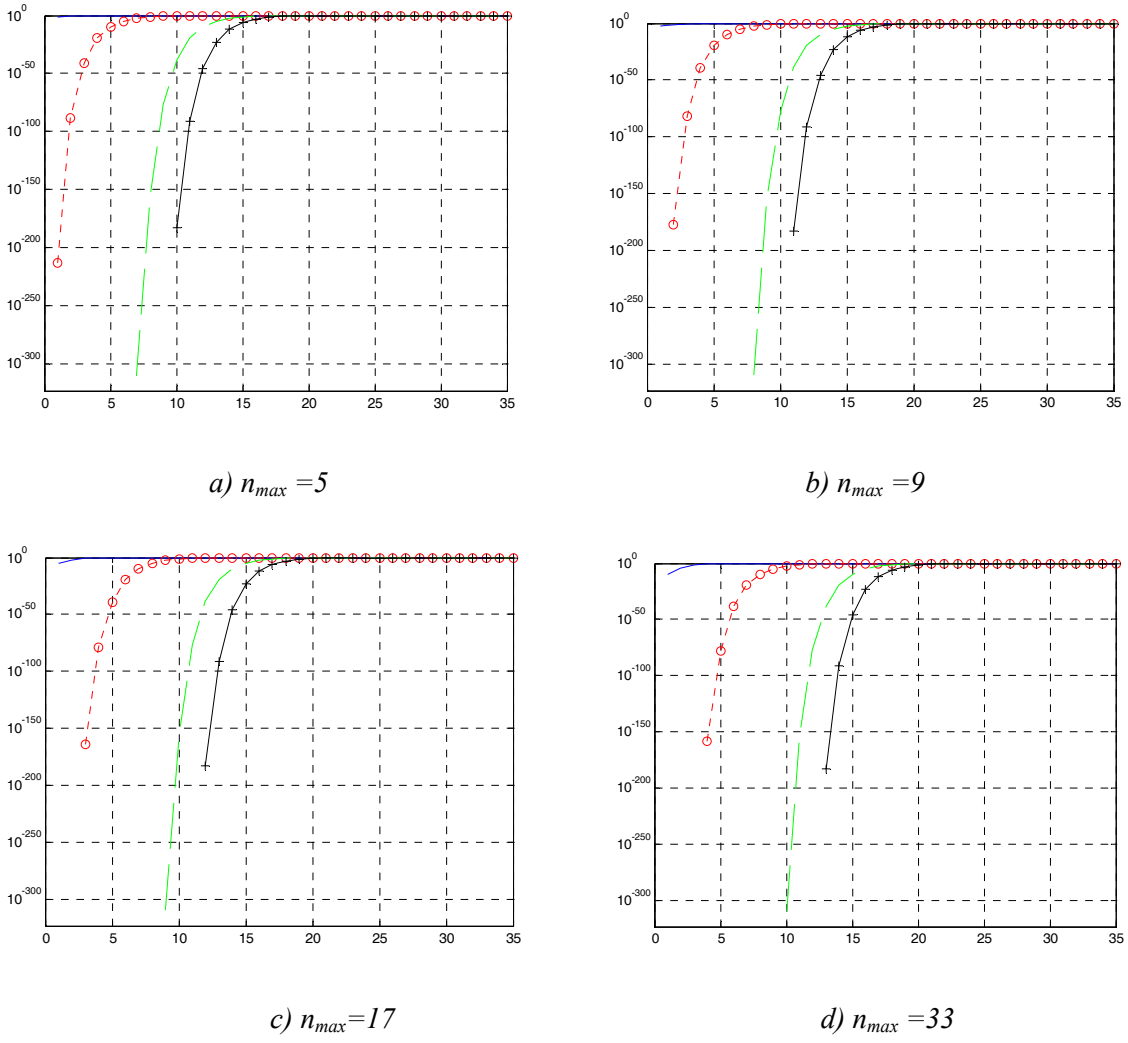
Figure 4.13 Probabilité de réussite de casser le système en fonction du nombre d'images tatouées utilisées.

Les résultats donnés par les différentes courbes nous amènent à faire les deux remarques suivantes :

1°- même si le nombre des coefficients à cryptanalyser augmente de façon très significative, le nombre d'images nécessaires pour casser l'algorithme varie très peu et ceci est vrai quel que soit la valeur de n_{max}.

2°- l'augmentation du n_{max} n'ajoute réellement rien pour augmenter la robustesse du système contre l'attaque. En effet, les quatre graphes montrent clairement que quel que soit la valeur de n_{max} utilisé, le nombre d'images nécessaires pour casser l'algorithme varie peu.

Par ailleurs, nous pouvons observer que les courbes obtenues ont la forme prédite dans la figure 4.11.



a) n_{max} *=5* *b)* n_{max} *=9*

c) n_{max}*=17* *d)* n_{max} *=33*

Figure 4.14. Probabilité de réussite de casser le système en fonction du nombre d'images tatouées utilisées.

La figure 4.14, utilisant une échelle logarithmique sur les axes des ordonnées, appelle les mêmes remarques que la figure 4.13, mais en plus, elle permet d'avoir une meilleure précision sur la valeur des probabilités pour les intervalles inférieurs de l'axe des abscisses.

Nous avons simulé et tracé les différentes courbes avec Matlab, et les valeurs des probabilités inférieures à 10^{-300} ne peuvent pas être représentées.

Pour compléter les informations fournies par les graphiques, nous avons calculé, utilisant la relation (4.13), la probabilité de cryptanalyser le système au moyen d'une seule image. Nous savons qu'il y a un nombre n_{max} de valeurs plausibles pour chaque coefficient au moment de l'utilisation de la première image. Une de ces valeurs est la valeur correcte et elle ne peut pas être éliminée. Chacune des autres valeurs au nombre $(n_{max}-1)$ ont une probabilité 0.5 d'être éliminée. Ces probabilités sont indépendantes entre elles et il faut que toutes les valeurs soient éliminées pour que la cryptanalyse soit réussie. Alors, la probabilité de trouver la valeur correcte du n_l, pour un coefficient donné, après avoir utilisé une seule image est donnée par :

$$P_{lim} = \frac{1}{2^{n_{\max}-1}} = 10^{-(n_{\max}-1)\log_{10} 2} \tag{4.13}$$

La probabilité de casser une zone d'image ou une image entière contenant L coefficients est alors $(P_{lim})^L$. Nous donnons dans le tableau 4.2 la probabilité de casser l'algorithme pour un seul coefficient, une région ou images composées de L coefficients et ceci en fonction des valeurs déjà choisies de n_{max}.

Tableau 4.2. Probabilité de casser l'algorithme en n'utilisant qu'une seule image

	Un coefficient AC	**Une région 70x70 pixels**	**Une image 700x600 pixelx**	**Une image 1160x1600 pixels**
n_{max} **=5**	$10^{-1,20}$	10^{-213}	10^{-27320}	$10^{-129609}$
n_{max} **=9**	$10^{-2,40}$	10^{-425}	10^{-54640}	$10^{-259219}$
n_{max} **=17**	$10^{-4,81}$	10^{-851}	$10^{-109509}$	$10^{-519518}$
n_{max} **=33**	$10^{-9,63}$	$10^{-1.704}$	$10^{-219246}$	$10^{-1040117}$

Nous pouvons observer pourquoi la majorité de ces valeurs ne peuvent pas être représentées sur le graphique de la figure 4.13.

Nous donnons dans les tableaux 4.3, 4.4, 4.5 et 4.6 le nombre d'images nécessaires permettant d'avoir une probabilité supérieure à des valeurs fixées, dans le cas d'un coefficient seul, une zone d'image 70x70 pixels, une image 720x600 pixels, et une image 1160x1600 pixels, et ceci pour les quatre valeurs utilisées de n_{max}. Ces données ont été tirées des courbes présentées dans les figures 4.13 et 4.14.

Tableau 4.3 : Nombre d'images requises pour casser un coefficient, une zone et images de taille différente pour une probabilité donnée, avec $n_{max}=5$

	Un coefficient AC	**Une région 70x70**	**Une image 720x600**	**Une image 1160x1600**
p=0.01	1	8	15	17
p=0.05	1	8	15	18
p=0.25	2	9	17	19
p=0.5	3	10	18	20
p=0.75	4	12	19	21
p=0.95	7	14	21	24
p=0.99	9	17	24	26

Tableau 4.4 : Nombre d'images requises pour casser un coefficient, une zone et images de taille différente pour une probabilité donnée, avec $n_{max}=9$

	Un coefficient AC	**Une région 70x70**	**Une image 700x600**	**Une image 1160x1600**
p=0.01	2	9	16	18
p=0.05	2	9	16	19
p=0.25	3	10	18	20
p=0.5	4	11	19	21
p=0.75	5	13	20	22
p=0.95	8	15	22	25
p=0.99	10	18	25	27

Tableau 4.5 : Nombre d'images requises pour casser un coefficient, une zone et images de taille différente pour une probabilité donnée, avec $n_{max}=17$

	Un coefficient AC	**Une région 70x70**	**Une image 720x600**	**Une image 1160x2048**
p=0.01	2	10	17	19
p=0.05	3	10	17	20
p=0.25	4	11	19	21

p=0.5	5	12	20	22
p=0.75	6	14	21	23
p=0.95	9	16	23	26
p=0.99	11	19	26	28

Tableau 4.6 : Nombre d'images requises pour casser un coefficient, une zone et images de taille différente pour une probabilité donnée, avec $n_{max}=33$

	Un coefficient AC	**Une région 70x70**	**Une image 700x600**	**Une image 1024x2048**
p=0.01	3	16	18	20
p=0.05	4	16	18	21
p=0.25	5	17	20	22
p=0.5	6	18	21	23
p=0.75	7	20	22	24
p=0.95	10	22	24	27
p=0.99	12	24	27	29

Dans les différents tableaux, nous pouvons remarquer qu'approximativement 9 images suffissent pour faire varier la probabilité de 0.01 à 0.99, et ceci quel que soit le nombre des coefficients traités et quel que soit la valeur de n_{max} utilisée.

En conclusion, dans cette partie, d'abord, nous avons cryptanalysé l'algorithme de Wang avec une attaque de type exhaustive sur les valeurs n_i. Ensuite, nous avons modélisé la cryptanalyse en utilisant la chaîne de Markov. Ceci nous a permis de montrer que le nombre d'images nécessaires pour mener avec succès la cryptanalyse est inférieur à 30 et il vari très peu avec la variation du nombre des coefficients tatoués ou avec le paramètre n_{max}.

4.5. Algorithme proposé (Caragata et al. 2010e)

Nous avons déjà montré les avantages de l'algorithme de Wang en termes de simplicité d'implémentation et d'intégration dans le standard JPEG. Cependant, nous avons démontré la fragilité de l'algorithme à l'attaque développée dans le paragraphe 4.3. Afin de pallier aux inconvénients de l'algorithme de Wang, mais de profiter de ses avantages, nous proposons un nouvel algorithme (Caragata, et al., 2010) caractérisé par :

1. Utilisation de générateurs chaotiques ayant de bonnes propriétés cryptographiques et donc très robustes.
2. Tatouage des coefficients DCT (DC et AC) qui ont une valeur supérieure à une certaine valeur-seuil *T*. Or, rappelons que l'algorithme de Wang tatouait tous les coefficients AC quantifiés plus grands que 1.
3. Utilisation d'une partie de la clé secrète K, pour cacher la valeur initiale du deuxième système chaotique. Ceci rend plus difficile la cryptanalyse de l'algorithme.
4. Rendre le nombre *n* d'itérations du premier système chaotique (sortie du système) variable pour chaque coefficient traité.

4.4.1. Générateurs chaotiques utilisés

Comme l'algorithme de Wang, l'algorithme proposé utilise deux générateurs chaotiques. Nous les avons choisis pour les puissantes propriétés cryptographiques qu'ils confèrent à l'algorithme. De cette façon, l'algorithme ne sera plus vulnérable à des attaques qui utilisent la faiblesse des signaux chaotiques.

Pour le premier système chaotique, nous proposons d'utiliser la fonction chaotique linéaire par morceaux (PWLCM – Piecewise Linear Chaotic Map). La carte PWLCM est composée de plusieurs segments linéaires par morceaux. Elle est donnée par l'équation suivante :

$$x_{n+1} = F(x(n)) = \begin{cases} x(n) \cdot \dfrac{1}{p}, 0 \le x(n) < p \\ [x(n) - p] \cdot \dfrac{1}{0,5 - p}, p \le x(n) < 0,5 \\ \dfrac{x(n) - 0,5}{p}, 0,5 \le x(n) < 0,5 + p \\ \dfrac{x(n) - 0,5 - p}{1 - 0,5 - p}, 0,5 + p \le x(n) < 1 \end{cases} \quad (4.14)$$

La clé secrète est composée de la valeur du paramètre p ($0 < p < 0.5$) et de la valeur initiale de la fonction, $x(0)$. Au total 64 bits pour une implémentation 32 bits.

Pour le deuxième système chaotique, nous proposons une structure composée de deux filtres récursifs montés en série et intégrant chacun une fonction non linéaire (El Assad et al, 2008). Dans ce travail, nous avons utilisé la fonction Skew Tent map comme fonction non linéaire. La figure 4.15 montre l'architecture du deuxième générateur :

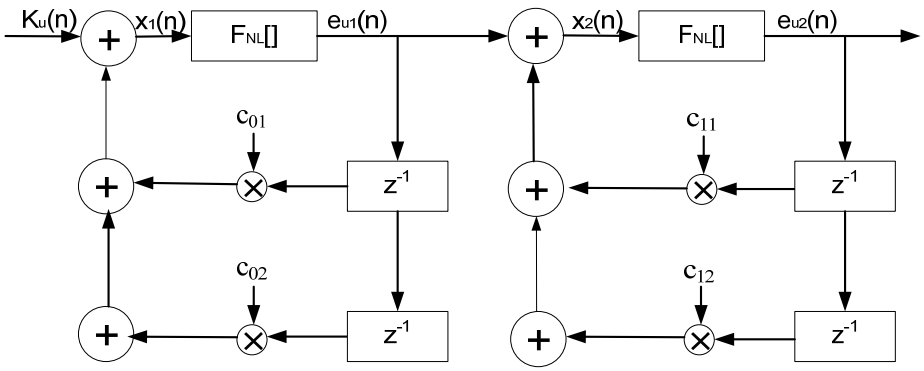


Figure 4.15. : Architecture du deuxième générateur chaotique proposé

Les équations à chaque niveau sont:

$$x_1(n)=k_u(n)+c_{01}\cdot e_{u1}(n-1)+c_{02}\cdot e_{u2}(n-2) \tag{4.15}$$

$$e_{u1}(n)=F_{NL}(x_1(n)); \tag{4.16}$$

$$x_2(n)= e_{u1}(n)+c_{11}\cdot e_{u2}(n-1)+c_{12}\cdot e_{u2}(n-2) \tag{4.17}$$

$$e_{u2}(n)=F_{NL}(x_2(n)); \tag{4.18}$$

Les coefficients c_{01}, c_{02}, c_{11}, c_{12}, sont des valeurs entières quelconques (sauf zéro) et font partie de la clé secrète. La fonction Skew Tent F :[0,1]→[0,1] est donnée par:

$$x(n+1)=F(x(n))=\begin{cases}\dfrac{x(n)}{p}, & \text{if } 0\le x(n)<p\\ \dfrac{1-x(n)}{1-p}, & \text{if } p\le x(n)\le 1\end{cases} \tag{4.19}$$

C'est une fonction qui transforme l'intervalle unitaire en lui-même. Elle dépend d'un paramètre p qu'on suppose satisfaire l'inégalité:

$$0<p<1 \tag{4.20}$$

La transformation est continue et linéaire par morceaux :

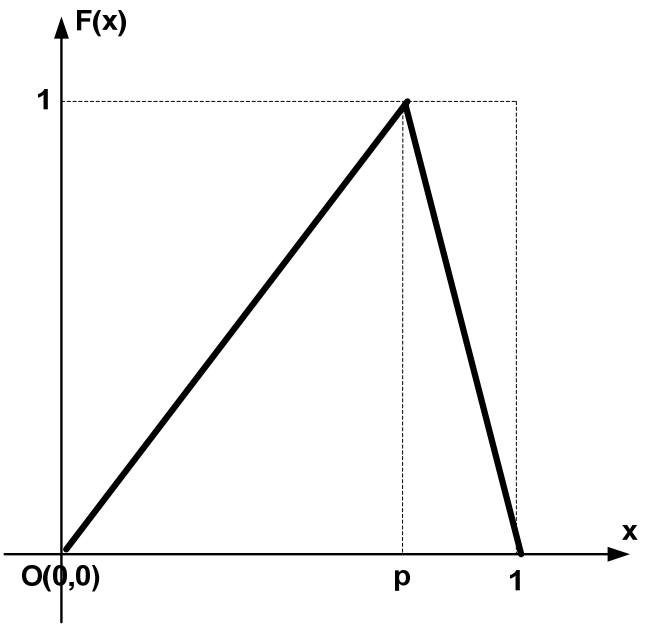


Figure 4.16. : Fonction chaotique Skew Tent map

La taille de la clé secrète du générateur proposé est très large, elle est composée de: paramètres c_{01}, c_{02}, c_{11}, c_{12} ; valeurs initiales $x_1(0)$ et $x_2(0)$, les deux paramètres de deux fonctions Skew Tent map et l'entrée K_u. Si chaque valeur est représentée sur 32 bits, le cas d'une implémentation typique, la taille de la clé secrète utilisée pour ce système chaotique est de 288 bits.

4.5.2. Procédure de tatouage des coefficients

L'algorithme de Wang insère le tatouage numérique sur tous les coefficients AC de la transformée DCT. Etant donné le nombre élevé de coefficients qui sont traités, l'algorithme est, d'une part, lent et, d'autre part, il apporte des modifications qui changent l'image d'une manière évidente. Pour pallier à ces inconvénients, nous proposons d'insérer le tatouage numérique sur tous les coefficients DC et AC ayant une valeur plus grande qu'un seuil fixé.

Les avantages de l'utilisation des coefficients DC comme porteurs d'information de tatouage ont été montrés par Huang (2000).

La technique de tatouage numérique proposée est basée sur la modification du bit LSB du coefficient sous traitement. C'est-à-dire, la valeur du coefficient traité est modifiée par 1. Si la valeur du coefficient est petite, par exemple 2, sa valeur peut devenir 3. Dans ce cas nous avons un changement de 50% de la valeur du coefficient. Or, dans un bloc de 64 coefficients, le nombre des coefficients AC ayant une petite valeur est important, et par conséquent, la procédure du tatouage implique des modifications visibles sur l'image résultat.

Nous savons par ailleurs que la valeur du coefficient DC par bloc est nettement plus grande que les autres valeurs des différents coefficients AC qui forment le bloc. Ceci a des conséquences sur le

résultat de l'image tatouée. En effet, considérons le cas d'un coefficient DC qui a une valeur de 60 (valeur de fréquence importante) et qui devient 61 après tatouage. Dans ce cas nous avons un changement inférieur à 2% de la valeur du coefficient et le bloc traité reste visiblement inchangé.

C'est pour cela que nous proposons l'ajout de l'information de tatouage dans les coefficients DC et dans les coefficients AC ayant une valeur plus grande qu'un seuil T donné.

Dans le paragraphe 4.5, nous allons montrer l'efficacité de notre approche en termes de distorsions minimales provoquées par la procédure du tatouage.

4.4.3. Utilisation d'une partie de la clé secrète pour le calcul de la valeur initiale y0

Dans l'algorithme de Wang, la valeur initiale du système chaotique 2 est obtenue en mettant simplement le LSB du coefficient DCT à zéro. Ainsi, la valeur initiale y_0 est connue par l'attaquant qui peut l'utiliser pour faciliter la cryptanalyse.

Dans l'algorithme proposé, la valeur initiale est rendue inconnue par l'utilisation d'une partie de la clé secrète comme suit :

$$y_0 = g_{ky0}(DCTi_{LSB=0}) \tag{4.21}$$

où y_0 est la valeur initiale du deuxième système chaotique, $DCTi_{LSB=0}$ est le coefficient sous analyse avec le LSB mis à zéro et k_{y0} constitue une partie égale à 32 bits de la clé secrète. De cette façon, l'attaquant n'aura pas accès à la valeur initiale de la séquence chaotique.

La fonction $g(\cdot)$ est donnée par :

$$g_{ky0}(DCTi_{LSB=0}) = (DCTi_{LSB=0} \cdot 2^{24} + k_{y0}) \bmod 2^{32}/2^{32} \tag{4.22}$$

Dans ce cas, les 32 bits de la clé k_{y0} sont interprétés comme un nombre entier et le résultat de cette fonction est un nombre entre 0 et 1 qui ne peut pas être calculé par l'attaquant sans la connaissance de k_{y0}.

4.5.4. Sortie du premier système chaotique (nombre *ns* des itérations) rendue variable en fonction de chaque coefficient traité

Pour les deux algorithmes (celui de Wang et le notre), l'information de tatouage pour chaque coefficient DCT est calculée en itérant le deuxième système chaotique n_i fois, où n_i est la sortie du premier système chaotique. Pour l'algorithme de Wang, les différentes sorties n_i, correspondantes aux différents coefficients à tatouer, sont obtenues en itérant une seule fois, pour chaque coefficient, le premier système chaotique. Notons que la valeur initiale du premier système chaotique à chaque

itération est celle obtenue par l'itération précédente. Alors, les mêmes valeurs n_i seront produites pour toutes les images tatouées avec la même clé. Ceci peut être employé par l'attaquant pour effectuer une attaque.

Afin de rendre l'algorithme résistant contre ce type d'attaque, nous proposons d'itérer la première fonction chaotique $ns1_i$ fois, avec :

$$ns1_i = h_{kn}(\mathrm{DCTi}_{\mathrm{LSB}=0}) \tag{4.23}$$

où $ns1_i$ est le nombre d'itérations à faire par le premier système chaotique, $\mathrm{DCTi}_{\mathrm{LSB}=0}$ est le coefficient sous analyse avec le LSB mis à zéro et k_n constitue une partie de la clé secrète sur 8 bits.

La fonction $h(\cdot)$ est donnée par :

$$h_{kn}(\mathrm{DCTi}_{\mathrm{LSB}=0}) = 1 + ((\mathrm{DCTi}_{\mathrm{LSB}=0} + k_n) \bmod 3) \tag{4.24}$$

Dans ce cas, la clé k_n est interprétée comme un nombre entier sur 8 bits et la sortie de la fonction $h(\cdot)$ prend aléatoirement les valeurs entières 1 ou 2 ou 3.

4.5.5. Architecture de l'algorithme proposé

Dans la figure 4.17 suivante, nous présentons l'architecture de l'algorithme proposé.

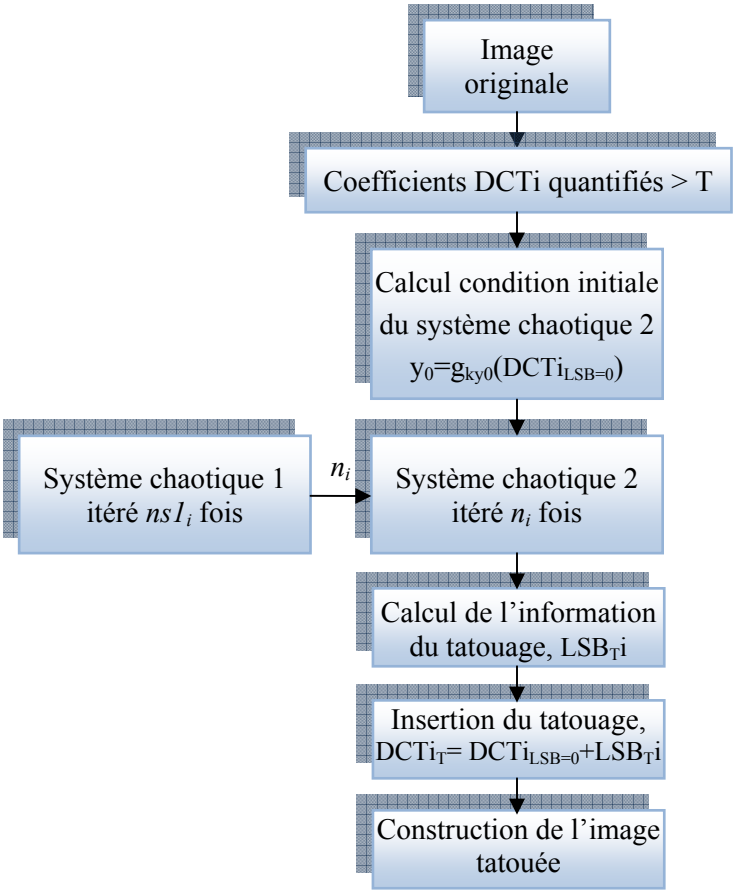


Figure 4.17. : Architecture de l'algorithme proposé

4.6. Résultats de simulation

Afin de montrer les performances en termes d'intégrité et d'imperceptibilité de l'algorithme de tatouage numérique proposé, en comparaison avec celui de Wang, nous avons implémenté sous Matlab les deux algorithmes en question.

D'abord, sur les figures 4.18 a), 4.18 b), 4.18 c) et 4.18 d) nous présentons respectivement : l'image originale 'Mountain.jpg' de taille 720×600 pixels ; l'image tatouée correspondante par notre algorithme de tatouage; l'image tatouée modifiée et l'image des régions altérées après vérification de l'intégrité de l'image tatouée par notre algorithme d'extraction.

a) Image originale : 'Mountain.jpg'

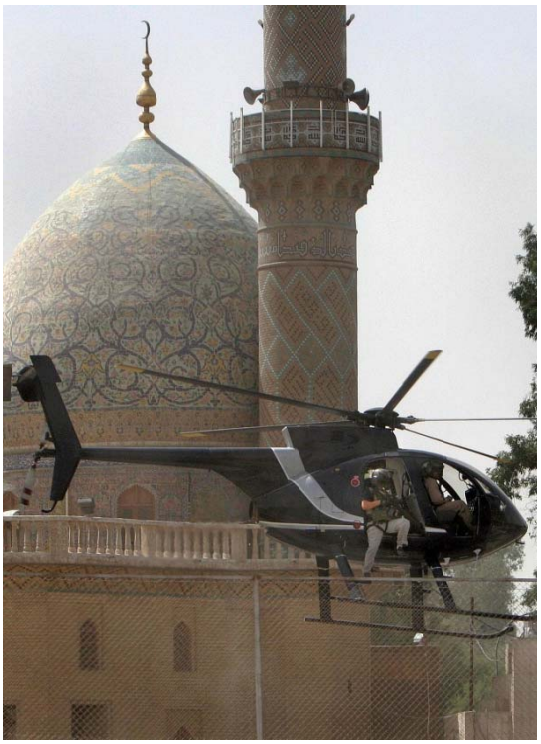
b) Image 'Mountain.jpg' *tatouée*

d) Image 'Mountain.jpg' tatouée et modifiée

c) Image des régions altérées

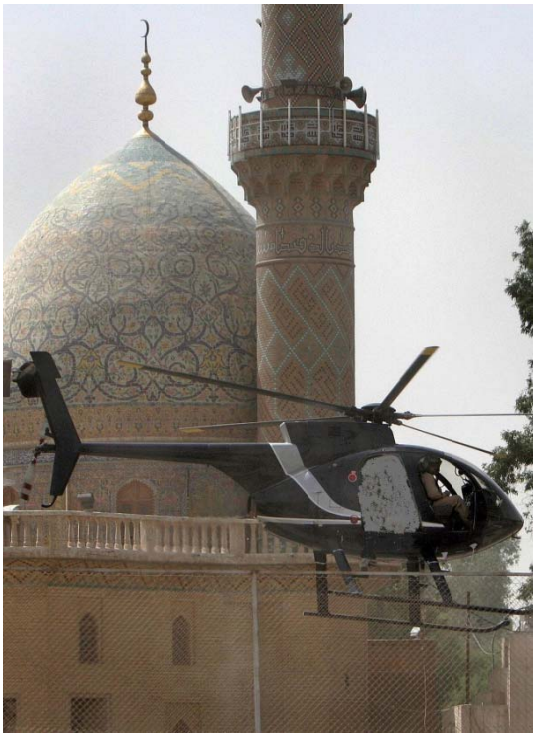
Figure 4.18. : Résultats de simulation sur l'image 'Mountain.jpg'

De même, sur les figures 4.19 a), 4.19 b), 4.19 c) et 4.19 d) nous présentons respectivement : l'image originale 'Hélicopter.jpg' de taille 1160x1600 pixels ; l'image tatouée correspondante par notre algorithme de tatouage; l'image tatouée modifiée et l'image des régions altérées après vérification de l'intégrité de l'image tatouée par notre algorithme d'extraction.

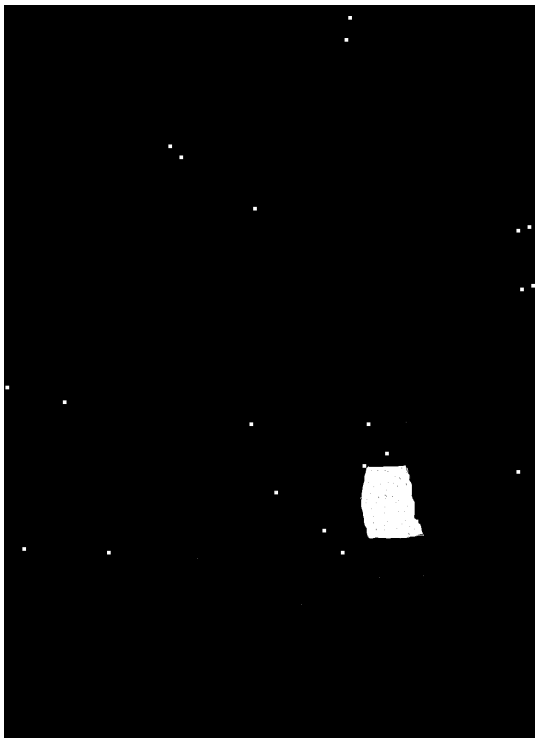


a) Image originale : 'Hélicopter.jpg'

b) Image 'Hélicopter.jpg' tatouée



c) Image 'Hélicopter.jpg' tatouée et modifiée



d) Image des régions altérées

Figures 4.19. : Résultats de simulation sur l'image 'Hélicopter.jpg'

Comme l'implémentation de l'algorithme est faite avec Matlab, la lecture d'une image par l'instruction « imread », retourne les composants RGB de l'image JPEG. L'écriture d'une image JPEG, avec l'instruction « imwrite » utilise les composantes RGB comme données d'entrée. Les deux instructions exécutent les étapes du standard JPEG, décrites dans la figure 4.2., sans qu'on puisse avoir

accès aux coefficients DCT quantifiés qu'elles utilisent. C'est pour cela que dans l'implémentation avec Matlab, après avoir lu l'image JPEG, nous devons calculer les coefficients DCT quantifiés en répétant les étapes du standard JPEG : découpage en blocs, transformation DCT, quantification. Une fois le tatouage inséré dans les coefficients DCT quantifiés, nous sommes obligés de calculer les 3 plans RGB qui seront utilisés par l'instruction « imwrite » afin de fournir le fichier JPEG. Ceci oblige aussi à répéter les étapes suivantes du standard JPEG : dé-quantification, transformation DCT inverse et reconstruction des plans RGB. Ceci introduit des erreurs, qui se manifestent sous forme de points blancs dans les images des régions altérées.

La figure 4.20 montre les étapes réalisées du processus du tatouage avec l'implémentation Matlab.

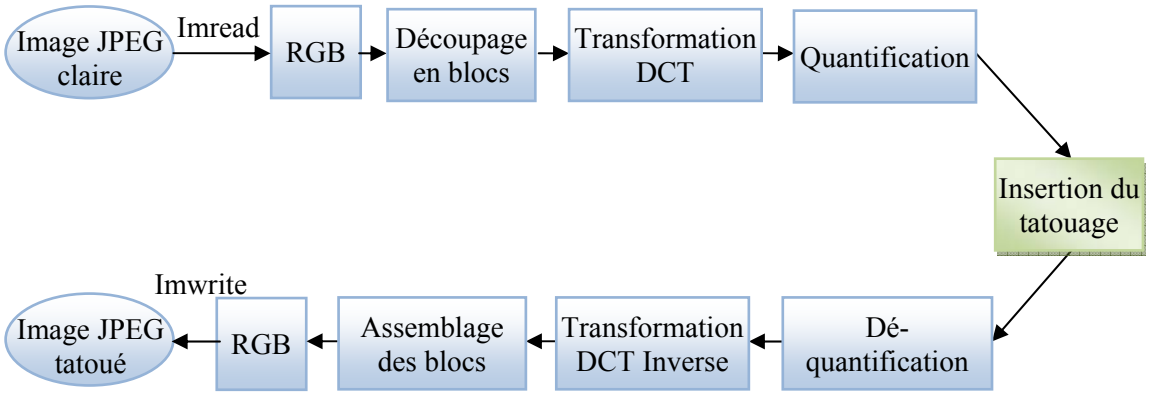


Figure 4.20. : Etapes du processus du tatouage implémentées avec Matlab

Nous allons maintenant démontrer l'efficacité de l'algorithme proposé par rapport à l'algorithme de Wang en termes d'imperceptibilité. A cet effet, nous utilisons deux mesures permettant de quantifier les changements amenés par le processus de tatouage numérique. Ces mesures sont : le PSNR (Peak Signal to Noise Ratio) et le coefficient de corrélation.

Le PSNR est un critère objectif de l'imperceptibilité qui mesure l'impact de l'insertion du tatouage « bruit » dans l'image originale. Il est défini par :

$$PSNR(f, f_w) = 10 \times \log_{10} \left[\frac{\max_{(m,n)} f^2(m,n)}{\frac{1}{M \times N} \sum_{(m,n)} (f_w(m,n) - f(m,n))^2} \right] \tag{4.25}$$

où f représente l'image originale, f_w l'image tatouée et (m,n) représente la location d'un pixel. Les valeurs typiques du PSNR, pour des images de bonne qualité visuelle, sont supérieures à 30 dB (Thomos, 2006).

De même, le coefficient d'intercorrélation, donné par la relation (4.27), peut être utilisé afin de mesurer la ressemblance entre l'image originale et l'image tatouée. Une valeur proche de 1 signifie une très bonne ressemblance.

$$E(f) = \frac{1}{M \times N} \sum_{(m,n)} f(m,n) \tag{4.26}$$

$$r(f, f_w) = \frac{\frac{1}{M \times N} \sum_{(m,n)} (f(m,n) - E(f))(f_w(m,n) - E(f_w))}{\sqrt{\frac{1}{M \times N} \sum_{(m,n)} (f(m,n) - E(f))^2} \sqrt{\frac{1}{M \times N} \sum_{(m,n)} (f_w(m,n) - E(f_w))^2}} \tag{4.27}$$

Dans les tableaux 4.7 et 4.8, nous avons donné les résultats comparatifs du PSNR et du coefficient d'intercorrélation, de l'algorithme de Wang et celui proposé, appliqués sur les images « *Mountain.jpg* » et « *Hélicopter.jpg* ». Nous pouvons observer que l'algorithme proposé est plus efficace avec un gain pratiquement de 2 dB.

Tableau 4.7 : Résultats comparatifs pour l'image « Mountain »

	Algorithme de Wang	**Algorithme proposé**
PSNR [dB]	77.5628	79.8531
Coefficient d'intercorrélation	0.9993	0.9995

Tableau 4.8 : Résultats comparatifs pour l'image « Hélicopter »

	Algorithme de Wang	**Algorithme proposé**
PSNR [dB]	80,0364	81,5942
Coefficient d'intercorrélation	0,9953	0,9966

4.7. Conclusions

Dans ce chapitre, nous avons proposé un algorithme de tatouage fragile, basé sur les signaux chaotiques, permettant de vérifier l'intégrité des images JPEG. Les performances de l'algorithme proposé semblent supérieures à celles de l'algorithme proposé par Wang et al en 2008, en termes d'imperceptibilité mesuré par le PSNR et la corrélation. Nous avons aussi conçu l'algorithme pour mieux résister aux attaques cryptographiques. Cette conception découle directement de la cryptanalyse de l'algorithme de Wang.

Par ailleurs, nous avons développé une méthode de cryptanalyse pour l'algorithme de Wang et nous l'avons modélisé en s'appuyant sur les chaînes de Markov du premier ordre. Cette méthode de

cryptanalyse est montrée efficace même si l'attaquant ne dispose que d'un faible nombre d'images tatouées (entre 10 et 30 images tatouées).

L'algorithme proposé est essentiellement utilisé pour assurer l'intégrité des images JPEG transmises, comme par exemple, l'intégrité des images lors de leur transfert sur l'Internet.

Pour des travaux futurs, nous proposons de concevoir des algorithmes qui permettront d'assurer l'intégrité des images mais aussi d'assurer la revendication de droit d'auteur (algorithmes de tatouages robustes). Nous proposons aussi de généraliser la technique de cryptanalyse développée afin de permettre l'évaluation de la sécurité des autres algorithmes de tatouage basé sur les signaux chaotiques.

Conclusion et perspectives

Conclusion et perspectives

Ces travaux de thèse ont permis, d'abord, et après une analyse approfondie de la question de la sécurité de l'information, de mettre en évidence les failles de sécurité qui se trouvent dans les systèmes de communications telles que: les communications IP par DVB satellitaire, les communications mobiles par réseaux UMTS et le tatouage fragile des images numériques. Ensuite, ils ont détaillé les différentes solutions proposées, basées sur les séquences chaotiques, pour contrecarrer leurs failles et apporter une nette amélioration de la question sécurité. Les différents services de la sécurité concernés par les solutions proposées sont: la confidentialité, l'authentification, l'intégrité et la non répudiation.

Dans le premier chapitre, nous avons introduit les ingrédients et les notions nécessaires pour la compréhension des sujets traités dans la thèse, comme l'InfoSec, le chaos, les protocoles et gestion des clés secrètes ou la sécurité des données multimédia.

Dans le deuxième chapitre, nous avons étudié la sécurité des communications IP par DVB satellitaire, dans le cas des communications unicast et multicast. Nous avons analysé d'abord, les procédures de sécurité utilisées dans les systèmes actuels des communications IP par DVB satellitaire et nous avons mis en évidence leurs faiblesses. Puis, nous avons proposé une solution de sécurité plus performante pour ce type des communications. La solution consiste à chiffrer les paquets IP transportés et le code MAC associés, protégeant ainsi l'authenticité de l'en-tête ULE et les paquets IP. Notre solution s'appuie sur une gestion des clés multicouche, des fonctions chaotiques pour la génération des clés et le chiffrement, un PDU spécifique pour le transport des clés et un message d'alarme pour rétablir la synchronisation entre le fournisseur et le client. Ensuite, nous avons quantifié les performances de cette solution en termes de taux des données ajoutées et nous avons montré sa supériorité par rapport aux autres solutions de sécurité existantes.

Dans le troisième chapitre, nous avons présenté et analysé la philosophie et l'architecture de sécurité des réseaux de communications mobile UMTS. De cette analyse, il s'avère que l'aspect le plus sensible de la sécurité UMTS est l'accès sécurisé au réseau. Nous avons présenté ces trois composantes : l'identification des utilisateurs, l'authentification, l'établissement des clés, via la procédure AKA, et l'établissement du chiffrement et de l'intégrité des données.

Ensuite, nous avons apporté des solutions, permettant d'améliorer la sécurité des communications mobile UMTS sur les trois aspects suivants: Le premier aspect concerne l'identification des utilisateurs, à ce propos des améliorations ont été supportés par rapport à la proposition d'Al Saraireh. Le deuxième aspect aborde la vulnérabilité de la clé secrète K, contre les attaques cryptographiques. Nous avons montré que sa vulnérabilité est comparable à celles des clés de

chiffrement CK et d'intégrité IK, mais que sa compromission affecte totalement toutes les communications passées et futures. Pour cela, nous avons protégé les messages de la procédure AKA et aussi renforcé la protection de la clé K, par un mécanisme inspiré de la procédure de protection de l'identité permanente IMSI. Le troisième aspect traite la question de la confiance dans le réseau de service. Afin d'améliorer la question de confiance, nous avons proposé des changements dans la procédure de choix des algorithmes de chiffrement et d'authenticité, et dans la procédure de changement du TMSI. Ceci permet à l'utilisateur et au réseau d'origine d'évaluer le comportement du réseau de service et de quantifier le degré de confiance qu'ils peuvent accorder à ce réseau.

Dans le quatrième chapitre, nous avons proposé un nouvel algorithme de tatouage numérique fragile, basé sur les séquences chaotiques, pour la vérification de l'intégrité des images JPEG. La conception de l'algorithme en question, découle directement de l'analyse de l'algorithme proposé par Wang en 2008. Suite à cette analyse, nous avons développé une méthode de cryptanalyse efficace qui a permis de casser l'algorithme de Wang. L'attaque en question, fait passer une image falsifiée pour une image non altérée après le processus de vérification de l'intégrité. Ensuite nous avons modélisé la procédure d'attaque, en s'appuyant sur les chaines de Markov. Nous avons aussi montré que, même lorsque le nombre des coefficients à casser augmente de façon très significative (de 1 à 108000), le nombre d'images nécessaires pour cryptanalyser est d'environ 20 images.

En perspective, pour les communications IP par DVB-S, nous allons approfondir l'étude de la sécurité dans le cas multicast et leur simulation par un simulateur DVB. Ensuite nous allons analyser la sécurité de nouveaux systèmes d'encapsulation des paquets IP pour DVB satellitaire tel le GSE (Generic Stream Encapsulation).

Pour le tatouage des images, basé sur les séquences chaotiques, nous proposons de concevoir un algorithme de tatouage permettant à la fois la protection des droits d'auteur et la vérification de l'intégrité. Aussi, nous proposons d'étendre la méthode de cryptanalyse que nous avons développée à d'autres algorithmes de tatouage basés chaos.

Annexes

Annexes

Annexe A1 : Communications satellitaires

Le premier satellite équipé d'un émetteur radio a été lancé le 4 octobre 1957 par les russes et s'appelait Spoutnik 1. Les américaines ont envoyé leur premier satellite de communications, Project SCORE, le 18 décembre 1958. Son premier message envoyé, fut le message du président Eisenhower pour Noël.

Les satellites de communications ont connu un développement continu et restent un moyen de communication vital parce qu'ils possèdent des caractéristiques intéressantes telles que: zone de couverture très vaste (échelle continentale), accès à des zones isolées (sommet d'une montagne, plateforme pétrolière, etc), à des zones sièges de catastrophes (tremblements de terre, incendie, inondation ou guerre) et à des objectives mobiles (avions, navires et trains).

Les satellites offrent une gamme variée de services de communications incluant: diffusion radio et de télévision, fax, téléphone, communications IP, etc.

Dans cette annexe, nous décrivons les types des satellites utilisés dans les communications en fonction de leur orbite et des services offerts.

A1.1. Orbites des satellites

Les satellites de communications peuvent être classés selon leur orbite:

- satellites en orbite géostationnaire
- satellites en orbite terrestre basse
- satellites en orbite Molniya.

A1.1.1. Satellites géostationnaires

L'inventeur supposé des satellites géostationnaires est Arthur C. Clarke, qui a écrit in 1945 l'article « Extra-terrestrial relays » dans le magazine britannique Wireless World, sur les lois fondamentales régissant le déploiement de satellites artificielles en orbite géostationnaire, à des fins de relayer les signaux radio. A cet effet, il s'est basé sur les travaux de Constantin Tsiolkovski et Herman Potocnik.

Le premier satellite de communications géostationnaire, Anik 1, a été lancé par le Canada le 9 novembre 1972 et restera en exploitation jusqu'au 15 juillet 1982.

Les satellites géostationnaires effectuent une orbite complète en 23 heures et 56 minutes, à une vitesse constante à la verticale de l'équateur. Alors, ils semblent fixes pour un observateur se trouvant à la surface de la Terre. Les antennes au sol doivent impérativement être pointées vers le satellite. Elles n'ont pas besoin de systèmes de poursuite de mouvement du satellite, qui sont très coûteux et

difficilement exploitable. Pour les applications ou le nombre d'antennes au sol est très grand (par exemple la diffusion des bouquets de télévision numérique) le coût de la mise sur une orbite haute (36 000 km) est justifié par les économies réalisées sur les équipements au sol.

Près de 40% des satellites en exploitation dans le monde entier ont été construits par Boeing Satellite Developement Center. Les autres fabricants importants sont Loral Space Systems, Lockheed Martin Space Systems, Northrop Grumman, Thales et EADS Astrium.

A1.1.2. Satellites en orbite terrestre basse

Les satellites en orbite terrestre basse sont placés à une altitude entre 350 et 1400 km de la surface de la Terre. Ils effectuent une orbite complète dans une période comprise entre 90 minutes et 2 heures. A cause de leur orbite basse, ils ont des caractéristiques tels que: un satellite est uniquement visible dans un rayon de quelques centaines de kilomètres et pour une duré de temps limitée. Alors, une constellation d'un grand nombre de satellites est nécessaire pour pouvoir offrir une connectivité permanente à une zone plus étendue. Ceci est possible, à cause du coût nettement inférieur au coût de la mise en orbite géostationnaire d'un satellite. Par exemple, le réseau Iridium utilise 66 satellites et le réseau Globalstar utilise 60 satellites.

Une autre utilisation possible de ces satellites est l'enregistrement des données reçues lors du passage au dessus d'une zone terrestre et la retransmission lors du passage sur une autre zone, comme c'est le cas pour le système Cascade du projet canadien Cassiope

A1.1.3. Satellites en orbite Molniya

Les satellites géostationnaires sont placés au-dessus de l'équateur, alors ils sont peu visibles sous des latitudes élevées et apparaissent très bas sur l'horizon. Ceci entraine une perturbation de la liaison par les couches basses de l'atmosphère.

Les satellites en orbite Molniya ont une orbite fortement inclinée par rapport à l'équateur (63.4°). Ils se caractérisent par un apogée de l'ordre de 40.000 km, dans l'hémisphère où ils doivent apporter leurs services, et un périgée de l'ordre de 1000 km dans l'autre hémisphère. Ces satellites ont une période de rotation de 12 heures, et ils peuvent offrir leurs services durant 4 heures dans chaque période. Alors, 3 satellites sont nécessaires pour offrir un service continu.

Le premier satellite Molniya a été lancé par la Russie en 1965 et a été utilisé pour des transmissions expérimentales de télévision en Sibérie et dans l'extrême orient russe. En 1967, un système de télévision, basé sur les satellites Moliya appelé Orbita, a été lancé. L'objectif essentiel du système en question était d'assurer le service de télévision pour les régions du nord russes.

D'autres systèmes existent aussi, comme le système de satellites de surveillance et de communications du DOD (Department of Defense) des Etats-Unis.

Les différentes orbites sont représentées dans la figure 1.1. La couleur noire représente l'orbite géostationnaire, en vert, l'orbite des satellites GPS, en cyan, les orbites des satellites en orbite basse et en violet l'orbite Molniya.

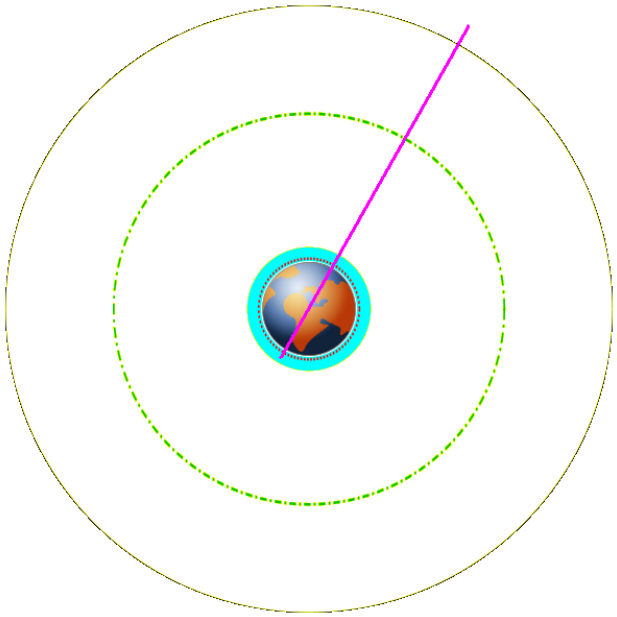


Figure A1.1: Orbites des satellites.

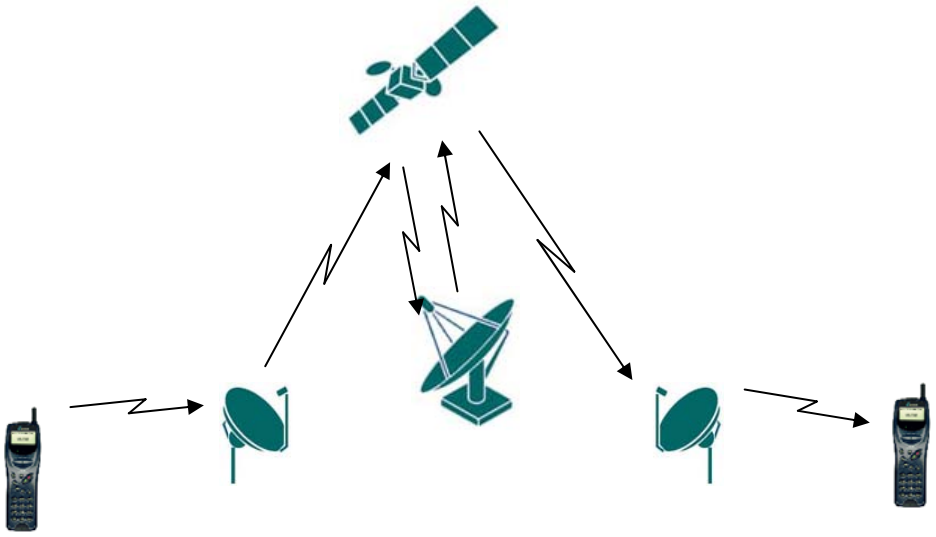
A1.2. Services de communications offerts

On peut distinguer les classes suivantes des services offerts par les satellites:

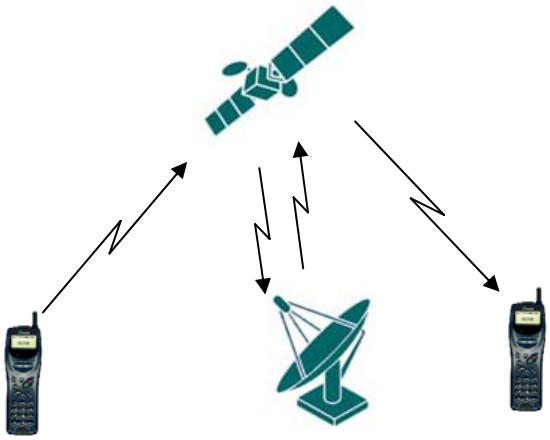
- téléphonie ;
- télévision et radio ;
- radioamateur ;
- Internet par satellite ;
- aide au système de positionnement ;
- satellites relais ;

A1.2.1. La téléphonie

La téléphonie internationale reste le service le plus important des satellites de communications. Selon le système de téléphonie, le terminal peut communiquer directement avec le satellite (pour les applications de téléphonie destinées aux navires, avions etc.) ou avec une station terrienne appelée aussi téléport, qui communique avec le satellite à sa place. Ensuite, le signal est dirigé par le satellite vers une autre station terrienne qui procède à la réception et à l'acheminent final. Les deux types de téléphonie satellitaire sont représentés dans la figure ci-dessous:



a) Téléphonie satellitaire utilisant une station terrienne intermédiaire



b) Téléphonie satellitaire sans station terrienne intermédiaire

Figure A1.2: Types de téléphonie satellitaire

A1.2.2. Télévision et radio

En télévision et radio, on distingue traditionnellement deux sortes d'applications:

- OU (Occasional Use): liaisons de contributions;
- ITV (International TV): diffusion de télévision.

ITV, est utilisé pour diffuser la télévision vers un nombre illimité d'utilisateurs qui possèdent des systèmes de réception le plus basique (pour l'Europe, des antennes de diamètres entre 0.6 et 1.1 mètres, placées directement chez les particuliers). En général, les fréquences utilisées sont en bande K. Les principaux fournisseurs sont BSkyB en Grand Bretagne, CanalSat en France, ExpressVu au Canada ou Sky Angel aux Etats-Unis.

OU, est utilisé pour avoir des images, qui ne se trouvent pas au siège d'une chaîne, afin de couvrir en direct un événement extérieur tel que: télémédecine, enseignement à distance,

visioconférences, etc. Alors, on est dans la situation d'une liaison de A vers B ou de A vers B, C, D…, avec un nombre limité de récepteurs.

A1.2.3. Internet et données par satellite

Lorsque les utilisateurs ne peuvent pas utiliser une connexion terrestre (ADSL, WiMAX, UMTS, réseau téléphonique, etc.) pour communiquer, alors, le transfert des données et l'accès à Internet par satellite est la solution adéquate. Aussi, ce type de communication permet aux entreprises implantées mondialement de ne pas dépendre des operateurs locaux de communications, qui ne sont pas toujours fiables et sont souvent sous contrôle des gouvernements locaux.

L'accès à Internet par satellite peut être réalisé soit avec une connexion satellitaire à deux sens (voie ascendante et voie descendante) soit avec une connexion satellitaire à sens unique (voie descendante) et un retour terrestre (dial-up, GPRS, etc.).

A1.2.4. Satellites relais

Les satellites relais sont utilisés pour permettre aux satellites en orbite basse (comme les satellites d'observation) de transmettre leurs données en temps réel vers la station terrienne. Les satellites en orbite basse envoient leurs signaux vers un satellite géostationnaire, qui les envoie à son tour vers la Terre. Les Etats-Unis utilisent quatre satellites relais pour une couverture globale.

Annexe A2: IP Sécurisé: IPSec

IPSec est un protocole au niveau réseau qui ajoute des services de sécurité au protocole IP (voir figure A2.1).

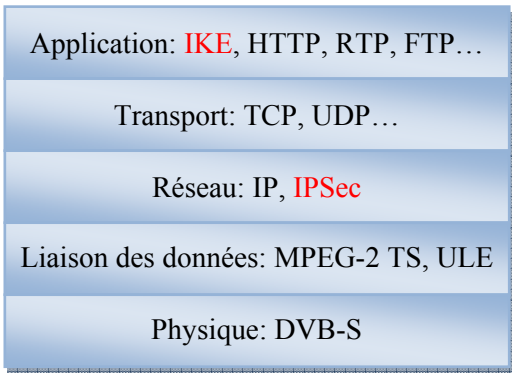


Figure A2.1: Modèle TCP/IP pour DVB-S sécurisé avec IPSec

Les services de sécurité offerts sont: la confidentialité, l'intégrité des données et l'authenticité de la source. Chaque paquet IP peut être soit authentifié, soit chiffré, soit les deux à la fois.

IPSec est, en effet, composé d'une série de protocoles. Les plus importants sont :

- *AH (Authentication Header)* – fournit l'authentification d'origine, et l'intégrité des données (IETF RFC 4302, 2005).
- *ESP (Encapsulating Security Payload)* – fournit la confidentialité, l'authentification d'origine et l'intégrité des données (IETF RFC 4303, 2005).
- *IKE (Internet Key Exchange)* – permet l'établissement d'une Association de Sécurité (SA – Security Association) par la négociation des clés, des protocoles et des algorithmes qui seront utilisés. IKE se base sur ISAKP (Internet Security Association Key Management Protocol) et les algorithmes Oakley et SKEME (IETF RFC 4306, 2005).

A2.1. SA (Security Association)

Une SA (IETF RFC 2401, 1998) est un concept de base pour IPSec. Une SA englobe l'ensemble d'algorithmes et des paramètres (e.g. clés, périodes de validités des clés) utilisés pour les services de sécurité. Une SA est unidirectionnelle, alors, pour sécuriser un trafic bidirectionnel, deux SA sont nécessaires, une pour chaque direction. La SA d'un certain paquet IP est définie par le SPI (Security Parameter Index), un champ de l'en-tête IPSec qui est un index pour la SADB (Security Association Data Base), et l'adresse destination. Plusieurs SA peuvent être utilisées, chacune avec son propre SPI, ce qui permet d'avoir plusieurs niveaux et ensemble de sécurité pour une certaine liaison.

A2.2. Modes: tunnel, transport

IPSec peut être utilisé en 2 modes différents :

- Mode tunnel.
- Mode transport.

En Mode tunnel, tous les paquets IP originaux sont chiffrés/authentifiés et un nouveau en-tête IP est créé pour chaque paquet. Ce mode est utilisé pour les VPN (Virtual Private Networks), il cache les caractéristiques du trafic (voir figure A2.2).

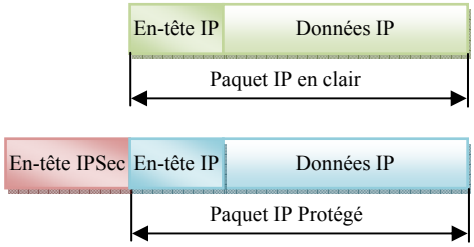


Figure A2.2: IPSec en mode tunnel

En mode transport, l'en-tête original IP est utilisé tel qu'il est et le chiffrement concerne seulement les données du paquet IP (voir figure ci-dessous)

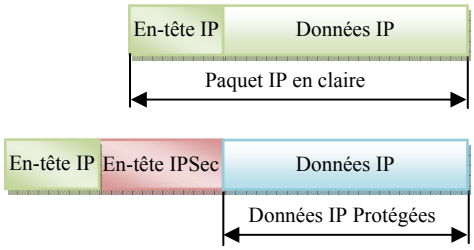


Figure A2.3. IPSec en mode transport

A2.3. Bases des données

IPSec utilise deux bases de données :

- SPD (Security Policy Database): contient les données qui permettent d'indiquer les services de sécurité requis pour chaque paquet IP traité par IPSec. Sa structure est décrite en (IETF RFC 4301, 2005).
- SAD (Security Association Database): contient tous les SA établies. SPD est la première base de données à être consultée. Si elle contient une SA qui satisfait les requis de la SPD, cette SA sera alors utilisée. Sinon, IKE est utilisé pour créer une nouvelle SA.

A2.4 IKE

La gestion des clés est réalisée avec un protocole spécifique, l'IKE. Il est utilisé pour créer les SA et les actualiser. Son fonctionnement est décrit dans la figure ci-dessous :

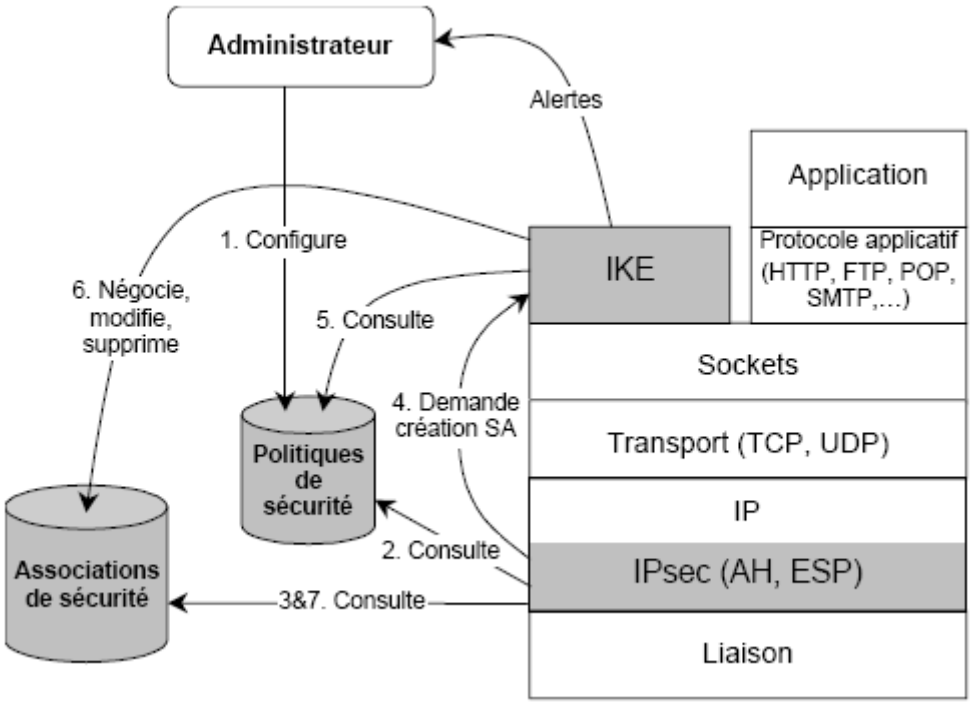


Figure A2.4: Protocole IKE

La première observation qu'on peut faire est que l'IKE est un protocole de niveau application, tandis que l'IPSec se situe au niveau réseau.

Pour que le système puisse fonctionner, l'administrateur doit, dans une première étape, configurer la SPD (1). Ensuite, pour chaque paquet IP qui doit être envoyé, IPSec consulte la SPD pour voir les services de sécurité requis pour cette connexion (2). Après, il vérifie s'il y a une SA dans le SAD qui supporte les requis (3). S'il n'y a pas, il envoie une demande à l'IKE (4). Celui-ci consulte le SPD (5), établie une nouvelle SA et l'ajoute au SAD (6). En fin, IPSec consulte à nouveau le SAD pour retirer la SA (7) et l'utilise pour appliquer les protocoles et algorithmes adéquats.

Les échanges des messages IKE sont indépendants d'IPSec et se déroulent en deux phases. La première phase permet l'établissement d'une SA propre à IKE. Contrairement aux SA d'IPSec, qui sont unidirectionnelles, cette SA est bidirectionnelle. Elle permet les échanges de la deuxième phase qui établissent les SA d'IPSec. Les messages de cette deuxième phase sont protégés par les protocoles, les algorithmes, et les clés de SA de l'IKE qui a été établie dans la première phase.

La première phase peut être exécutée en deux modes: *mode principal (main mode*) et *mode agressif (aggressive mode)*. Le mode principal, consiste en 6 messages. Le mode agressif compacte l'information qui sera échangée en 3 messages seulement.

IKE utilise le protocole ISAKMP. Celui-ci a été conçu pour être souple et pour être utilisé par d'autres applications. Ses messages sont constitués d'un chaînage des blocs. Il existe 13 types de blocs possibles:

- *SA – Security Association* : indique le contexte de l'échange en première ou deuxième phase, ainsi le fait qu'il s'agit d'un échange IPSec.
- *P – Proposal* : contient des propositions pour le SA : mécanisme à utiliser (AH ou ESP), et SPI à associer au SA.
- *T – Transform* : est un bloc complément au bloc *Proposal* et contient des propositions sur les algorithmes de chiffrement ou fonctions de hachage à utiliser pour la SA.
- *KE – Key Exchange* : transporte les données nécessaires à la génération de la clé de session.
- *ID – Identification* : transporte les données utilisées pour l'identification des tiers.
- *CERT – Certificate* : utilisé pour le transport des certificats (s'ils sont utilisés)
- *CR – Certificate Request* : demande un certificat.
- *HASH – Hash* : résultat d'une fonction de hachage.
- *SIG – Signature* : résultat d'une fonction de hachage signée.
- *NONCE – Nonces* : transporte des aléas (des données utilisées qu'une seule fois)
- *N – Notification* : permet l'échange des messages d'erreur ou d'information sûr l'état actuel des négociations.
- *D – Delete* : spécifie qu'une SA sera effacée.
- *VID – Vendor ID* : permet à deux installations de même marque de se reconnaître pour pouvoir utiliser des implémentations propres.

Annexe A3: Gestion de la clé

A3.1 Introduction

Les principes de la cryptographie, qui ont été présentés par Auguste Kerckhoffs en 1886 dans son livre de référence « La cryptographie militaire », (Kerckhoffs A., 1883), restent encore valables plus de 100 ans après. Un aspect très important qui a été souligné est que le secret de l'information réside dans la clé secrète et non dans l'algorithme de chiffrement. En effet, l'algorithme de chiffrement peut tomber dans les mains de l'ennemi, des cryptanalistes, sans que la confidentialité des données soit affectée. Cela veut dire qu'un bon algorithme de chiffrement transfère toute la confidentialité ou l'authenticité du message chiffré à la clé de l'algorithme. Seulement, celui qui connait la clé peut déchiffrer le message chiffré. On peut voir à partir d'ici que, après avoir choisi une machine de chiffrement robuste, le domaine de gestion de la clé secrète est primordial.

A3.2. Techniques de gestion des clés

A3.2.1. Notions de base

L'établissement des clés est un processus ou protocole qui permet à deux ou plusieurs entités de partager un secret commun qui va être utilisé dans des applications cryptographiques.

Le processus d'établissement des clés peut être divisé en deux tâches «*transport des clés»* et *«traitement des clés»*.

Le transport des clés est un protocole ou technique qui permet à une entité d'obtenir ou de créer une valeur secrète et de la transférer d'une manière sûre à une autre entité.

Le traitement des clés (ou négociation des clés) est un protocole ou technique qui permet à deux ou plusieurs entités de calculer une valeur secrète commune utilisant des données propres aux entités en question. Aucune autre entité non autorisée ne peut pas retrouver par calcul la valeur secrète en question. Les deux processus (transport et traitement des clés) sont utilisés dans les systèmes cryptographiques symétriques et asymétriques.

Changer la clé d'un algorithme de chiffrement le plus souvent possible permet d'augmenter la sécurité de communication d'une manière importante. En effet, cela permet de limiter la quantité de texte crypté avec une seule clé et ainsi de limiter la quantité d'information compromise si un attaquant réussit à casser le code. C'est pour cela, que la plupart des protocoles ont pour objectif la création des clés différentes pour chaque exécution. Un tel protocole est appelé protocole *dynamique*. Une clé qui a été générée par un tel protocole est appelée *clé de session*. Par conséquence, un tel protocole est aussi appelé protocole *de traitement de la clé de session*.

On a aussi des cas où pour chaque exécution du protocole de traitement des clés, on obtient la même clé. Un tel protocole est appelé protocole *de pré-distribution des clés* et il est vulnérable aux *attaques à clés connues*.

Une *attaque à clés connues* est une attaque où un adversaire dispose de clés de sessions antérieures, ce qui lui permet soit de calculer les clés présentes et futures, soit d'imiter une des parties impliquées dans le protocole.

Les protocoles de transport des clés sont des protocoles *dynamiques* alors que les protocoles de traitement des clés peuvent être des protocoles *dynamiques* ou des protocoles *de pré-distribution des clés*.

Pour qu'un agresseur ne soit pas capable d'avoir accès aux clés résultantes, il est généralement souhaitable que toutes les parties d'un protocole d'établissement des clés soient capables de déterminer les identités des autres. L'authentification dans la gestion de la clé a pour objectif la connaissance de l'identité de l'entité qui a accès à la clé.

L'authentification des clés est la procédure par laquelle une entité est assurée qu'aucune autre entité, à part une entité particulière identifiée, n'a accès à certaines clés secrètes. *L'authentification des clés* est réduite seulement à l'assurance que la clé n'est pas accessible aux entités non autorisées, elle est aussi appelée authentification *implicite des clés*.

Confirmation des clés: est la propriété par laquelle une entité est assurée qu'une autre entité (qui peut être non identifiée) est en possession d'une clé particulière.

Authentification explicite des clés: est une propriété obtenue lorsqu' on a l'authentification implicite des clés et la confirmation des clés.

La différence entre l'authentification *des clés* et *la confirmation des clés* est que la première se focalise sur l'identité de l'entité qui demande la clé, tandis que la deuxième se focalise sur la valeur de la clé reçue par l'entité.

En général, la confirmation des clés est réalisée par un processus qui indique qu'une entité reçoit un message d'une autre entité possédant la preuve que celle -ci est en possession de la clé. En pratique, la possession de la clé peut être prouvée par d'autres moyens, comme l'exécution d'une fonction hash à sens unique sur la clé même, ou le cryptage avec la clé d'un message connu.

L'authentification de l'entité n'est pas exigée dans tous les protocoles. Quelques protocoles de traitement des clés n'utilisent ni l'authentification de l'entité, ni l'authentification de la clé, ni la confirmation de la clé. On peut toujours ajouter une confirmation des clés unilatérale, c. à. d, en employant une fonction hash à sens unique.

Pour un protocole d'établissement des clés, qui implique l'utilisation d'un protocole d'authentification de l'entité, il est très important de s'assurer que l'entité pour laquelle on vérifie l'identité est la même que celle pour laquelle on a établi la clé. Sinon, un adversaire peut se servir de

l'authentification d'une entité autorisée et ainsi de pouvoir imiter cette entité pour le protocole de traitement des clés.

A3.2.2. Tiers de confiance

Nombre de protocoles d'établissement des clés utilisent un tiers de confiance, soit pour des actions d'initialisation du système, soit pour des actions on-line, soit pour les deux. Selon les fonctions qu'il exécute, ce tiers peut avoir des noms et des fonctionnements très variés.

Si on tient compte de leurs interactions en temps réel avec les autres entités du système, on peut avoir de tiers "*in line*", "*on line*" ou "*off line*". On appelle un tiers "*in line*" si la communication toute entière entre les entités impliquées dans un protocole d'établissement des clés est faite à travers le tiers de confidence. On appelle un tiers "*on line*" s'il est directement impliqué dans le protocole d'établissement des clés, mais les entités du protocole peuvent communiquer entre elles sur des canaux directs, sans qu'elles impliquent le tiers. On appelle un tiers "*off line*" si sa contribution au protocole d'établissement des clés est faite a priori en fournissant l'information qui va être utilisée ultérieurement. A noter aussi, que le tiers off line ne sera pas du tout impliqué quand le protocole d'établissement des clés se déroule entre les entités.

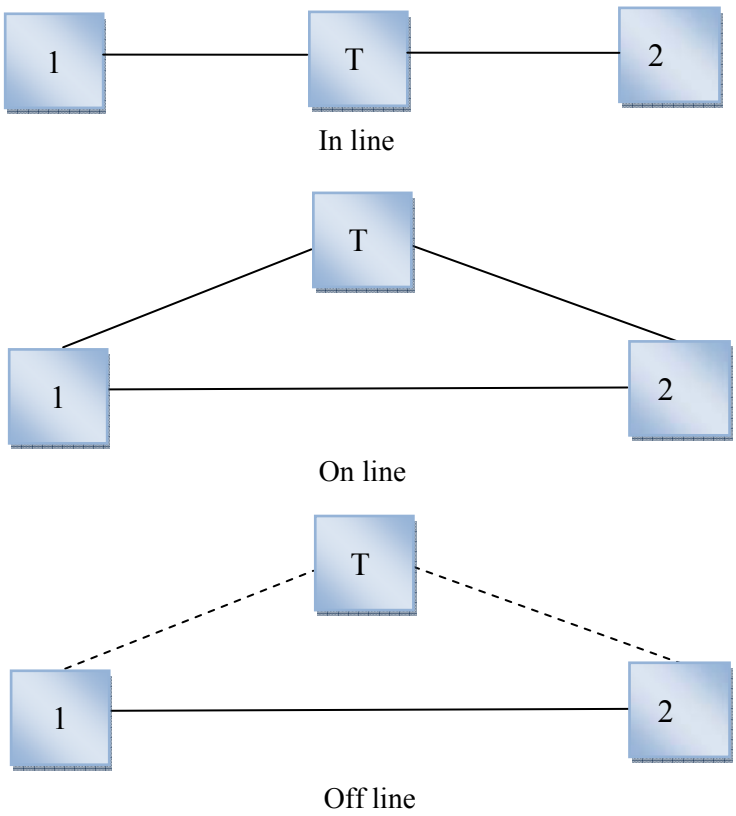
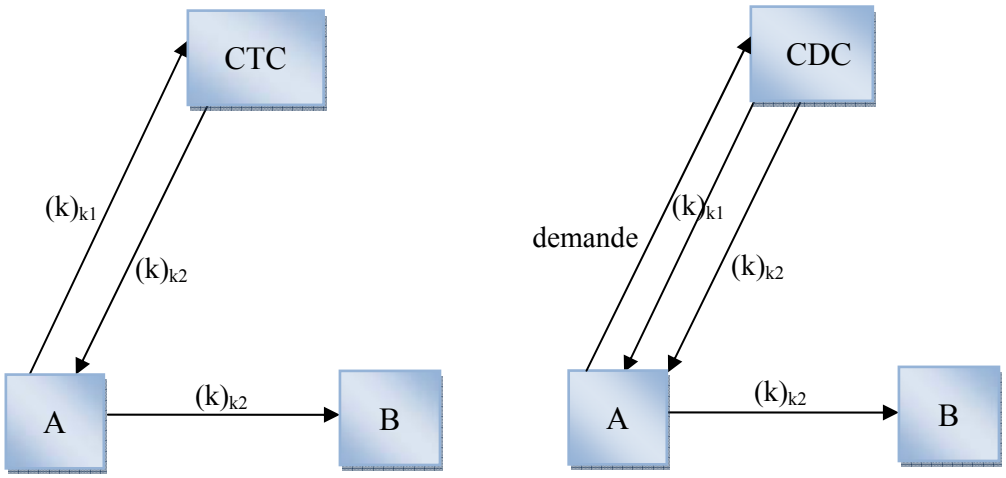


Figure A2.1. : Types d'interactions d'un tiers de confiance

Si on tient compte du type de services que le tiers fournisse aux participants, on peut avoir soit un CDC, Centre de Distribution des Clés (KDC – Key Distribution Center), soit un CTC, Centre de Transfert des Clés (KTC – Key Translation Center).

Le CTC, utilise des clés partagées a priori avec chacun des participants, impliqués dans le protocole afin de permettre une transmission sécurisée des clés sur des canaux non protégés. En d'autres mots, le CTC partage a priori une clé secrète avec chaque participant. Il reçoit une clé cryptée d'un des participants *A*, il décrypte la clé en question (en utilisant la clé partagée), et la crypte avec les clés des participants avec qui *A* veut communiquer en sécurité. Il envoie ces messages à *A* qui, à son tour les envoie aux autres participants sur un canal ouvert, car chaque destinataire ne peut décrypter que le message crypté avec sa propre clé.

La seule différence entre un CTC et un CDC est que le CDC ne reçoit pas la clé d'un des participants, mais il génère la clé lui-même. C'est-à-dire, le participant *A*, qui initialise le protocole de distribution des clés, n'envoie pas une clé au CDC mais une demande. C'est le CDC qui génère la clé. Ensuite, le CDC envoie aux participants impliqués la clé cryptée avec la clé partagée a priori entre le CDC et chaque participant.



Fonctionnement d'un CTC

Fonctionnement d'un CDC

Figure A3.2: Fonctionnement du CDC et du CTC

A3.2.3. Couches des clés

Parfois, un protocole d'établissement des clés implique l'utilisation de plusieurs clés, chacune avec un rôle différent. On peut classifier ces clés selon leur rôle:

- *Clé maître* – MK (Master Key): la clé de plus haut niveau, qui n'est pas protégée avec des mécanismes cryptographiques, mais avec des mécanismes physiques: elle est distribuée manuellement ou installée à la création du système et elle est protégée avec des mécanismes physiques, procédurales ou d'isolation électronique.

- *Clés de niveau intermédiaire* – IK, (Intermediary Keys) : clés de niveau intermédiaire qui sont utilisées pour le transport ou pour le stockage d'autres clés. On peut avoir ici plusieurs sous-niveaux de clés.
- *Clé de Session* – SK, (Session Key) : la clé de plus bas niveau, utilisée pour les opérations cryptographiques désirées: chiffrement, authentification, etc. Normalement les clés de session ont une très courte durée de vie.

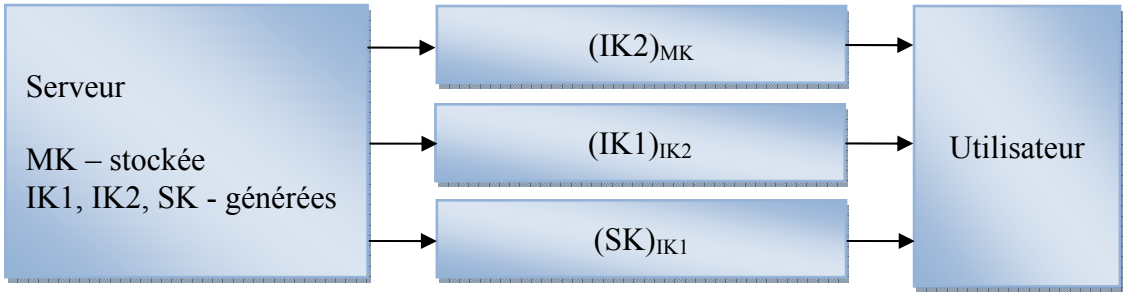


Figure A3.3: Fonctionnement d'un système de gestion de clé qui utilise plusieurs couches de clés

En regardant la figure A3.3, on peut se demander quel est le rôle des IK. Si on envoie la SK cryptée avec le MK nous obtenons un protocole qui utilise moins de ressources. Les clés IK sont nécessaires car elles nous permettent de satisfaire deux besoins contradictoires. D'un côté, on veut changer la clé SK le plus suivant possible, pour qu'un cryptanalyste ne puisse pas avoir assez d'information cryptée avec une seule clé, ou pour limiter la quantité des données compromises, quand la clé SK est compromise. D'un autre côté, on veut utiliser la clé MK le plus rarement possible, par ce qu'on ne peut pas la changer. Si elle est compromise, tout le système de gestion des clés est compromis et, par conséquence, tout le système de communication sécurisé est compromis. La solution qui satisfait les deux besoins contradictoires (changer SK sans utiliser MK) est d'utiliser un nombre des clés intermédiaires. Les clés de plus haut niveau sont, normalement, des clés à longue durée et les clés des niveaux inférieurs sont des clés à courte durée.

A3.2.4. Diffie Hellman

Diffie-Hellman est le premier algorithme à clé publique qui fut inventé. Sa sécurité dépend de la difficulté de calculer des logarithmes discrets sur un corps fini par rapport à la facilité de calcul d'exponentielles dans le même corps. Diffie-Hellman peut être utilisé pour la distribution de clés, mais il ne peut pas être utilisé pour chiffrer et déchiffrer des messages. Alice et Bob peuvent utiliser cet algorithme pour engendrer une clé secrète.

Les mathématiques sont simples. Au départ, Alice et Bob se mettent d'accord sur deux grands entiers, n et g, de telle manière que g soit primitif par rapport à n. Ces deux entiers ne doivent pas être secrets; Alice et Bob peuvent convenir de ces nombres sur un canal non sécurisé. Ils peuvent même être communs à un groupe d'utilisateurs. Peu importe.

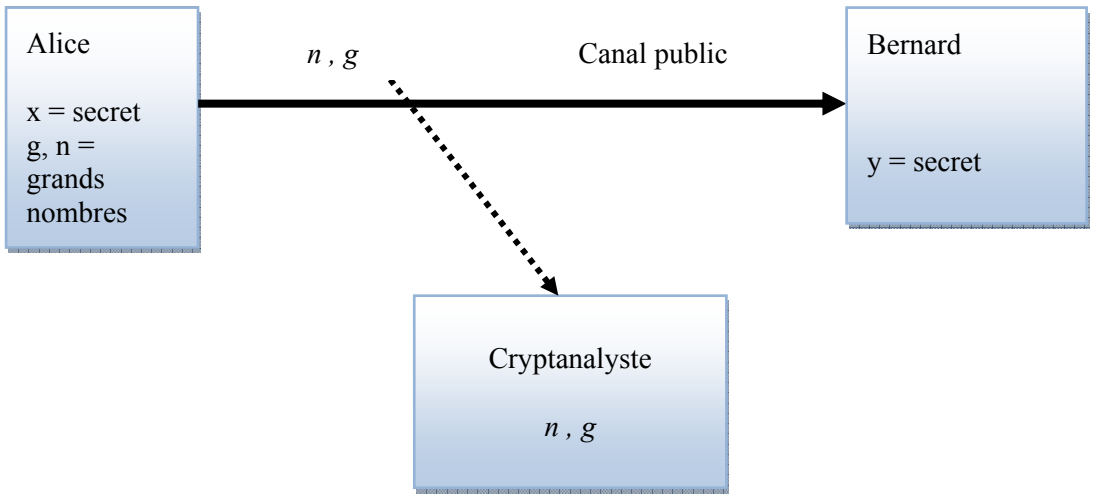


Figure A3.4: Initialisation de Diffie-Hellman

Le protocole se déroule ainsi:

- Alice choisit un grand nombre entier aléatoire x et envoie à Bernard le résultat du calcul: $X= g^x \bmod n$.
- Bernard choisit un grand nombre entier aléatoire y et envoie à Alice le résultat du calcul: $Y= g^Y \bmod n$.
- Alice calcule: $k= Y^X \bmod n$.
- Bernard calcule: $k'= X^Y \bmod n$.

Les valeurs *k* et *k'* sont toutes deux égales à $g^{xy} \bmod n$. Personne ne peut, en écoutant la communication, calculer cette valeur; celui qui écoute ne connaît que *n, g, X* et *Y*. À moins qu'il ne puisse calculer le logarithme discret et retrouver *x* ou *y*. Ainsi *k* est la clé secrète qu'Alice et Bernard ont calculée indépendamment.

Alice	Canal public	Bernard
Première étape		
choix du secret x		choix du secret y
traitement des n et g.	n, g	traitement des n et g.
Deuxième étape		
envoie	$m_{12} = g^x \bmod n$	réception
réception	$m_{21} = g^y \bmod n$	envoie
Troisième étape		
calcul: $k_1 = (m_{21})^x \bmod n =$ $g^{x*y} \bmod n = k$		calcul: $k_1 = (m_{12})^{x2} \bmod n =$ $g^{x*y} \bmod n = k$
Traitement des clés fini		

Figure A3.5: Diffie-Hellman avec deux utilisateurs.

Le choix de g et n peut avoir un impact substantiel sur la sécurité de ce système. Le nombre $(n-1)/2$ doit être premier. Plus important, n doit être grand, la sécurité du système dépend de la difficulté de factoriser des nombres de taille de n. Et g doit être n'importe quelle racine primitive modulo n; rien n'empêche de prendre la plus petite valeur de g qui convient (c'est généralement un nombre à un chiffre).

A3.3. Aspects de sécurité

A3.3.1. Période de vie des clés

Nous avons déjà dit, qu'après l'établissement des clés, une clé ne peut pas rester statique, sauf pour MK. Plusieurs raisons incitent à changer les clés le plus souvent possible:

- si un cryptanaliste possède assez de textes cryptés avec une seule clé, il aura plus de possibilité à trouver la clé de chiffrement;
- si une clé est compromise, nous voulons que la quantité d'information cryptée avec cette clé soit très petite;
- chaque algorithme de chiffrement peut crypter une quantité limitée d'information avec une seule clé;
- la clé SK peut être accessible pour un cryptanalyste, car elle peut se trouver dans une mémoire ou dans un endroit qui n'est pas sécurisé. C'est le cas du système de communication 3G, où

la clé est utilisée par l'équipement mobile. C'est facile pour un cryptanaliste de lire la mémoire d'un tel équipement;

Les clés doivent être changées avant l'expiration de la cryptopériode. Le changement implique l'utilisation du matériel pour les clés existantes dans des protocoles appropriés d'établissement des clés. Pour limiter la vulnérabilité du système en cas de compromission de certaines clés, il est souhaitable qu'il y ait une très grande indépendance entre les différents types de matériel pour les clés. Par exemple, on ne doit pas calculer une nouvelle clé à partir d'une ancienne.

Nous allons discuter par la suite, les étapes qui mènent au changement des clés. Dans (Menezes A., 1996) les auteurs présentent un système montrant toutes les étapes de la période de vie de la clé. C'est le plus complexe. La plupart des protocoles pratiques n'ont qu'une partie de ces étapes:

- *Enregistrement de l'utilisateur* – étape dans laquelle une entité devient un membre habilité d'un domaine sécurisé. Il s'agit de l'acquisition ou de la création et l'échange de matériel initial pour les clés, c. à. d, partage des mots de passe ou codes PIN, par une technique sécurisée (par exemple la carte à puce, l'échange personnel, courrier de confiance etc.).
- *Initialisation de l'utilisateur* - étape dans laquelle une entité initialise une application cryptographique (par exemple, installation et initialisation logicielles ou matérielles). Cela implique l'utilisation ou l'installation du matériel initial pour les clés obtenues au cours de l'enregistrement de l'utilisateur.
- *Génération de clés* – la génération de clés cryptographiques doit être faite de manière visant à assurer des propriétés appropriées pour l'application ou l'algorithme. Une entité peut générer ses propres clés, ou acquérir des clés d'un système de confiance.
- *Installation des clés* – le matériel pour les clés est installé pour une utilisation opérationnelle au sein d'une entité utilisant différentes techniques, par exemple: entrée manuelle du mot de passe, utilisation d'un PIN ou d'autres matériels.
- *Enregistrement de la clé* - en association avec l'installation des clés, la saisie, des données caractérisant les clés, peut être enregistrée, par une autorité d'enregistrement, et associée à un nom unique qui distingue une entité. Pour les clés publiques, des certificats des clés (public key certificate) peuvent être créés par une autorité de certification et mis à disposition par un annuaire public ou d'autres moyens.
- *Utilisation normale* - l'objectif du cycle de vie des clés est de faciliter leurs disponibilités opérationnelles. Dans des circonstances normales, cet état se poursuit jusqu'à l'expiration de la cryptopériode.
- *Sauvegarde de la clé* - sauvegarde des clés pour la récupération éventuelle des clés.
- *Mise à jour des clés* - avant la date d'expiration de la cryptopériode, le matériel opérationnel est remplacé par du nouveau matériel. Ceci peut impliquer la génération des clés, la

dérivation des clés ou la communication avec un tiers de confiance. Pour les clés publiques, la mise à jour et l'enregistrement de nouvelles clés se traduisent généralement par des protocoles de communication sécurisés avec les autorités de certification;

- *Destruction des clés* - une fois la clé est jugée obsolète, elle est retirée de tous les documents officiels, et toutes les copies de la clé sont détruites.
- *Récupération des clés* - si la clé est perdue d'une manière non compromettante (par exemple, en raison de défaillance de l'équipement ou des mots de passe oubliés), il peut être possible de la restaurer avec une copie de sauvegarde.
- *Révocation de la clé* – lorsque la clé est compromise, elle est immédiatement supprimée avant l'expiration de sa durée de vie initialement prévue.

A3.3.2. Problèmes de sécurité de protocoles cryptographiques

La gestion de la clé est une partie intégrante d'un système de communication sécurisé. Dans un système de communication sécurisé, plusieurs fonctions peuvent être demandées telles que:

- un message soit transmis seulement à une destination prévue,
- un expéditeur d'un message ne peut pas nier ses actions,
- un message ne soit pas altéré au cours de sa transmission, etc.

Dans tous ces cas, on suppose l'existence d'un adversaire.

Dans ce paragraphe, nous allons discuter les problèmes de la gestion des clés, en nous concentrant d'une part, sur l'adversaire et ses modalités d'attaque et d'autre part, sur les attaques auxquelles les protocoles d'établissement de la clé peuvent être soumis.

Appelons par participant, une entité communicante dans un protocole d'établissement de la clé et possédant un identificateur unique. Supposons aussi, la présence d'un tiers non autorisé qui a des noms variés, selon les circonstances: *adversaire*, *intrus*, *opposant*, *ennemi*, *agresseur*, *imitateur*.

Normalement, on fait une distinction entre les adversaires, selon leur niveau d'accès à l'information utilisée dans le protocole. On a deux types d'adversaires: *outsider* et *initié*. Un *outsider* est un adversaire qui a accès seulement à l'information disponible généralement. C'est à dire à l'information qui circule dans les canaux publics utilisés par le protocole. Un *initié* a normalement accès à des informations supplémentaires, comme des clés de session passées, obtenues par une attaque pratique, par exemple.

Les stratégies qu'un adversaire peut suivre sont variées. Il peut essayer de deviner la clé de session en utilisant une certaine information obtenue par la surveillance de la communication. Il peut aussi essayer de participer à un protocole initié par un participant pour influencer le résultat de la session. Enfin, il peut initier plusieurs fois le protocole, lui permettant de combiner les messages obtenus, afin d'imiter un participant, etc.

Quand on examine la sécurité d'un protocole d'établissement de la clé, on suppose que les mécanismes cryptographiques utilisés (i.e. algorithmes de cryptage, ou procédures de signature numérique) sont sûrs. Sinon le protocole n'est pas sûr.

On part du principe, que l'adversaire peut attaquer aussi la manière dans laquelle les mécanismes cryptographiques employés par le protocole (exemple: fonction de hachage) sont utilisés ensemble.

Généralement, on part du principe que les messages du protocole sont transmis sur des réseaux non sécurisés et contrôlés par un adversaire qui est capable d'enregistrer, de changer, d'effacer, d'insérer, de rediriger, de réorganiser, ou de réutiliser des messages. Pour mieux souligner le cadre incertain dans lequel le protocole est utilisé, on suppose toujours que les principaux échanges de messages passent d'abord par des adversaires susceptibles de prendre des actions hostiles de types cités plus haut sans que les participants soient alertés. On suppose aussi qu'un adversaire peut initier avec un participant le lancement d'un protocole

Une *attaque passive* suppose qu'un adversaire ne fasse qu'enregistrer et analyser de manière approfondie les données. Une *attaque active* suppose qu'un adversaire puisse aussi insérer, modifier ou altérer des messages échangés au cours du protocole.

A3.3.3. Planifications de contingences

Les classes d'utilisateurs des systèmes de communications sécurisées sont très variées. On peut avoir des utilisateurs privés; des systèmes de vente (vidéo à la demande, vente de musique on-line, etc.); des systèmes de communications dans les hôpitaux, des systèmes de communications bancaires, des systèmes de communications gouvernementaux, etc.

Pour chacun des ces systèmes, la perte des clés (ou leur compromission) a des conséquences différentes: pour un utilisateur privé, cela peut impliquer la perte des données personnelles (e.g. des photos), pour un système de payement à la demande, la perte de la transmission d'un match de foot. Mais pour certains systèmes, les dégâts provoqués peuvent être plus lourds: un patient peut mourir parce que le docteur n'a pas accès à ses analyses, un système gouvernemental peut être bloqué etc.

On va prendre comme référence pour la gestion des clés les recommandations du NIST (National Institute of Standards and Technology), États Unis (Barker E., 2006). La planification des contingences doit tenir compte des plans et mesures pour une très grande diversité de situations, telles que:

- perte des badges des clés ou des jetons;
- oublie d'un mot de passe qui donne l'accès aux clés secrètes;
- panne des dispositifs de lecture des clés;
- perte ou corruption des dispositifs mémorisant des clés;

- coupure du courant, impliquant la réinitialisation du système ou de certaines étapes concernant les algorithmes d'établissement ou de transport des clés;
- perte, corruption ou confusion entre l'association entités-clés;
- indisponibilité des anciennes versions des logiciels ou des équipements nécessaires pour avoir accès aux clés ou à l'information protégée.

Un aspect très important de la planification des contingences est l'accès aux données. Autres problèmes qui doivent être pris en compte sont : la restauration d'accès (sans perdre les restrictions, dans des endroits où les mécanismes cryptographiques sont utilisés), ou la restauration des processus utilisant des mécanismes cryptographiques d'autorisation, sans perdre les restrictions établies.

Si certaines clés utilisées pour protéger l'information ou des processus sensibles sont dévoilés, les clés compromises doivent être révoquées et remplacées, puis une opération d'estimation des dégâts doit être effectuée. Si les clés étaient partagées par plusieurs entités, les dégâts sont très étendus et le processus de restauration de la sécurité devient complexe et coûteux.

Annexe B : Description du fonctionnement du réseau UMTS

Dans cette annexe, nous présentons le standard UMTS. La compréhension de la structure du réseau est indispensable à toute analyse de la sécurité. Le réseau UMTS peut être structuré en:

- Equipement mobile : l'ensemble des composants qui sont chez l'utilisateur.
- Réseau d'accès : la partie du réseau UMTS qui gère la connexion radio avec l'équipement mobile.
- Réseau cœur : la partie du réseau UMTS qui contient les basses des données UMTS et les interfaces avec d'autres réseaux.

B.1. Equipment mobile

Le marché des terminaux mobiles a connu un développement croissant ces dernières années. Au début des années 90 les terminaux étaient mono-bande (900 MHz ou 1800 MHz) et supportaient seulement les services de la téléphonie et les messages courts SMS (Short Message Service). Au fil du temps, les téléphones sont devenus multi-bande: support des données haut débit EDGE (Enhanced Data Rates for GSM Evolution), support des applications paquets GPRS (General Packet Radio Service), support multi-mode : UMTS, GSM, WIFI, DVB-H, support des applications circuit et paquets haut débit HSDPA (High-Speed Downlink Packet Access), support pour l'LTE (Long Terme Evolution). Dans la plupart des cas, les avances technologiques ont été réalisées, en général et particulièrement dans le cas de l'UMTS, d'une manière qu'il ait une compatibilité entre les nouveaux standards et ceux en place. Une exception à cette règle est le Japon, ou la technologie UMTS a été mise en place par l'opérateur NTT DoCoMo sans qu'il ait une compatibilité avec le réseau de deuxième génération utilisé.

B.1.1. Structure de l'équipement utilisateur

La structure du terminal mobile UMTS est évoluée par rapport terminal mobile GSM. Dans le cas du GSM la station mobile MS (Mobile Station) est composée d'un équipement mobile ME (Mobile Equipment) doté d'une carte à puce. Dans le cas du mobile 3G (UMTS, etc…), l'équipement utilisateur UE (User Equipment), a une structure plus complexe. Il divise le ME, en MT (Mobile Termination) et en TE (Terminal Equipment). L'ensemble ME et la carte à puce USIM (Universal Subscriber Identity Module) qui contienne une application spéciale, est l'UE. La carte à puce de l'UMTS est évoluée par rapport a celle du GSM. L'accès aux services dans un réseau UMTS est conditionné par la présence de la carte à puce. Sans elle, le ME ne peut pas accéder aux services. Il peut seulement appeler les numéros d'urgence (112 en Europe, 911 aux États-Unis). La structure de l'UE est montrée dans la figure B.1.

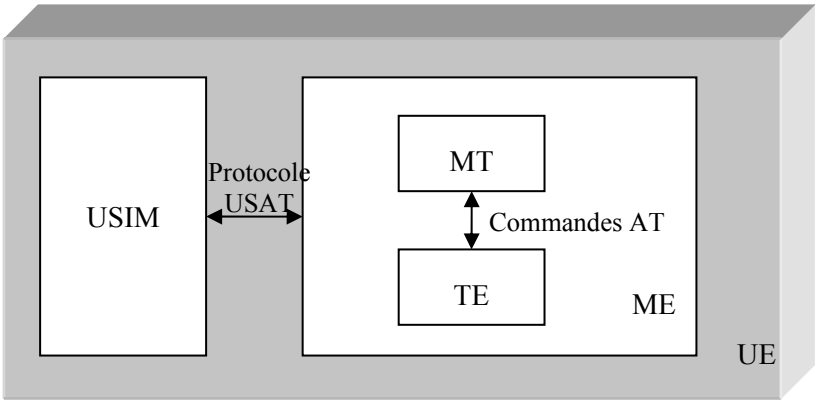


Figure B.1: Structure de l'équipement d'utilisateur UE en UMTS

Le MT gère la communication avec l'interface radio UTRAN (UMTS Terrestrial Radio Access Network). Il gère les fonctions de la couche physique ainsi que les protocoles de niveau 2. Le TE travaille au niveau application, il englobe les services directement accessibles à l'usager. L'interaction entre le MT et le TE est réalisée avec une interface qui utilise des commandes AT (3GPP TS 27.007).

B.1.2. Carte à puce USIM

La carte à puce utilisée en GSM et en UMTS est appelée UICC (Universal Integrated Circuit Card). Pour les réseaux GSM, la carte à puce contient l'application de sécurité SIM. Pour l'UMTS l'application est évoluée et est appelée USIM. Les spécifications de l'UICC pour le GSM et l'UMTS sont précisées dans les standards 7816 de l'ISO/IEC. Les différents aspects traités dans les spécifications ISO sont : les caractéristiques physiques, les dimensions et positions des contacts électriques, les signaux électroniques et protocoles d'échanges des données avec la carte, le système de fichiers et les commandes applicatives.

Actuellement il existe deux formats de carte à puce:

- Le format « plug in », 15x25 mm, qui est le plus courant, appelé aussi ID-000.
- Le format « mini », 12x15 mm, introduit à la demande du Japon pour permettre la réduction de la taille des terminaux. Il est appelé aussi mini UICC.

Toutes les cartes contiennent 8 contacts. Deux sont réservés et les autres sont:

- VCC : Alimentation.
- GND : Masse.
- RST: Réinitialisation.
- VP : Reprogrammation.
- CLK: Horloge.
- I/O: Echange des données.

La carte UICC, comme la plupart des cartes à puce, intègre les éléments suivants:

- Un microprocesseur.
- Une mémoire ROM (Read Only Memory) contenant le système de gestion de fichiers et les données permanentes.
- Une mémoire EEPROM (Electrically Erasable Programable Read Only Memory) contenant les données persistantes.
- Une mémoire RAM (Random Access Memory) contenant les données temporaires.

L'alimentation des cartes peut être réalisée avec une tension de 5 V, de 3 V ou de 1.8 V. Toutes avec une précision de 10%. La fréquence de l'horloge peut varier entre 1 et 5 MHz. La communication avec la carte à puce est décrite par les spécifications ISO/IEC 7816-3 et est réalisée en mode *half-duplex*.

Les données contenues dans la carte UICC sont structurés en fichiers distincts. La gamme des données contenues dans une carte est très variée. Pour l'application USIM par exemple, comme types de données, nous avons: l'IMSI, le numéro d'appel, le répertoire, la clé secrète permanente *K*, les clés temporaires de chiffrement et d'authentification *CK* et *IK*, la liste des réseaux interdits, l'identité temporaire de l'utilisateur TMSI, etc. La norme ISO 7816-4 décrit la structure des fichiers de la carte. Il s'agit d'une structure arborescente qui contienne trois types de fichiers :

- MF: (Master File), est le fichier maître, situé à la racine de la structure arborescente.
- EF: (Elementary File), est un fichier de base utilisé pour stocker l'information. Il existe trois types de format pour les EF:
  - Le format transparent – une suite d'octets non structurés.
  - Le format fixe linéaire (linear fixed) – enregistrements de taille fixe, structurés. Un exemple est le fichier répertoire.
  - Le format cyclique – utilisé, par exemple, pour le journal d'appels.
- DF: (Dedicated File), est un répertoire qui regroupe un nombre des fichiers EF. Il s'agit du même concept utilisé par les dossiers des ordinateurs.

Chaque fichier a des règles très strictes pour la lecture, l'écriture et la mise a jour des données. La norme 3GPP décrit quatre types de conditions d'accès:

- ALW: (Always), indique que l'information est accessible sans restriction. Elle est utilisée pour les données les moins sensibles, e.g. la langue utilisée.
- PIN: (Personal Identification Number), indique que l'information est accessible seulement après la vérification du PIN de l'usager.
- ADM: (Administrative), indique que seul le fournisseur de la carte peut accéder aux informations.

- NEV: (Never), indique que l'information n'est pas accessible.

Par exemple, le fichier EF_{IMSI}, qui contient l'IMSI de l'abonné, utilise la règle PIN pour la lecture et ADM pour la mise à jour. De même, les fichiers EF_{Keys} et EF_{KeysPS}, qui contiennent les clés de chiffrement et d'intégrité pour le domaine circuit, respectivement paquets, ont la règle PIN pour la lecture et pour la mise à jour.

Le standard 3GPP considère que l'UICC doit pouvoir contenir plusieurs applications: plusieurs USIM ou USIM et d'autres applications de type: carte bancaire, porte-monnaie électronique, etc. L'utilisation des plusieurs USIM sur la même UICC est contrainte par la recommandation qu'une seule carte USIM est active à un moment donné.

Les autres applications possibles dans le domaine des télécommunications sont: l'ISIM (IP Multimedia Services Identity Module), les réseaux IP multimédia et l'EAP (Extensible Authentication Protocol) utilisé pour l'accès aux réseaux Wi-Fi. L'ISIM, (TS 31.103) est l'équivalent de l'USIM pour un réseau IP multimédia. Il a été conçu comme une application indépendante pour les terminaux dédiés uniquement à ce type de communications. Les operateurs de télécommunications mobiles s'opposent actuellement à une telle approche, car cela permettra aux operateurs fournissant uniquement des services IP de profiter de leurs réseaux de service radio. L'EAP (TS 102.310) permet à un terminal Wi-Fi d'offrir des services d'authentification renforcés, basés sur les algorithmes 3G présents sur la carte. Cela est particulièrement intéressant pour les operateurs qui offrent des services de téléphonie et aussi des points d'accès à l'Internet.

La carte à puce n'est pas l'esclave du terminal mobile, elle est un élément proactif, capable d'envoyer des commandes pour provoquer des affichages, récupérer des informations ou échanger des données à travers le terminal mobile. L'environnement qui permet cette interaction (TS 31.111) est l'USAT (USIM Application Toolkit) et est composé d'un jeu de 34 commandes optionnelles incluant:

- DISPLAY TEXT: Affichage du texte ou d'une icône sur l'écran du terminal.
- SETUP CALL: Etablissement d'un appel circuit.
- SEND SHORT MESSAGE: Envoie du SMS à travers le terminal.
- RUN AT COMMAND: Envoie d'une commande AT de la carte USIM vers le MT comme si elle était issue du TE.

B.2. Réseau d'accès radio

L'UTRAN est le réseau d'accès qui assure la liaison entre les terminaux mobiles et le réseau cœur UMTS (voir figure B.3). Il est considéré comme une version évoluée du réseau d'accès GSM représenté dans la figure B.2.

B.2.1. Réseau d'accès GSM

En GSM, le réseau d'accès est appelé BSS (Base Station Subsystem) et comprend deux types d'éléments:

- BTS: (Base Transceiver Station), éléments constitués d'émetteurs/récepteurs avec une intelligence très limitée.
- BSC: (Base Station Controller), contrôle un ensemble des BTS.

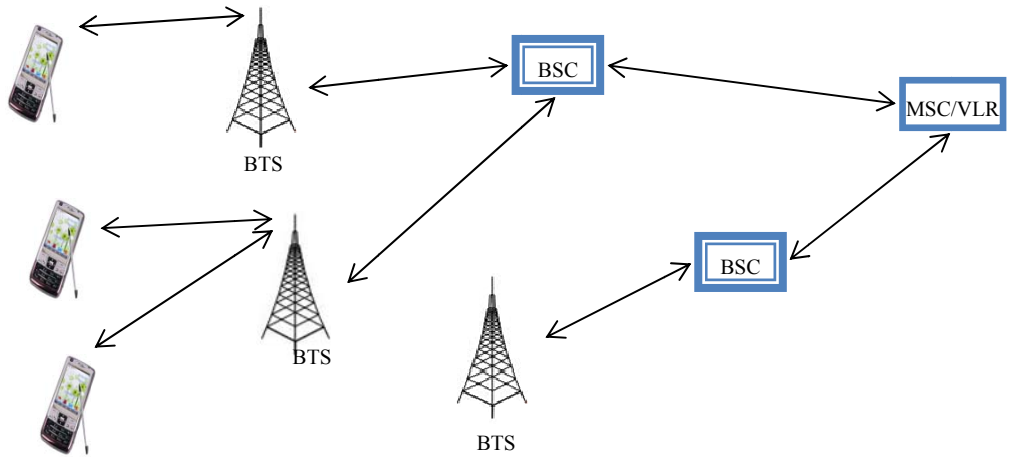


Figure B.2: Architecture du BSS

Le BTS est un ensemble d'émetteurs/récepteurs appelé TRX. Il gère toute la couche physique (modulation, démodulation, égalisation, codage correcteur d'erreurs, multiplexage TDMA, saut de fréquence lent, etc.) et la couche liaison des données pour l'échange de la signalisation entre les mobiles et l'infrastructure. Il réalise également, l'ensemble des mesures radio nécessaires pour vérifier qu'une communication en cours se déroule correctement.

Les operateurs emploient massivement la tri-sectorisation: c'est-à-dire, placer sur le même site trois BTS contenant chacun des antennes directionnelles qui couvrent 120°. Les trois BTS sont généralementplacés dans une même armoire et apparaissent comme un seul équipement. La norme GSM voit les trois BTS comme des équipements différents, tandis que le vocabulaire courant parle d'un « BTS tri-sectorisé ». Ces ambigüités de vocabulaire sont levées avec l'UMTS ou la norme reprend le vocabulaire courant (voir paragraphe *Nœud B* du B.2.2).

Le BSC est l'organe intelligent du BSS et il a pour fonction principale le management de la ressource radio: il commande l'allocation des canaux, utilise les mesures effectuées par la BTS pour contrôler les puissances d'émission des mobiles et de la BTS, prend la décision d'effectuer un handover, etc. Il réalise aussi le relayage des circuits vers le MSC (Mobile Switching Center).

B.2.2. Eléments du réseau UTRAN

Le réseau d'accès UMTS, l'UTRAN, a une structure similaire au BSS, il est constitué des stations de base et de contrôleurs, voir figure B.3. Les différences avec le BSS sont quand même importantes et ont poussé à un changement de vocabulaire:

- Le Nœud B (Node B): est un ensemble d'émetteurs/récepteurs qui correspond à la BTS du GSM.
- Le RNC (Radio Network Center): est le contrôleur des ressources radio et correspond au BSC du GSM.
- Le réseau de transport: est une nouveauté par rapport au GSM et il assure la liaison entre les différents éléments du réseau UTRAN.

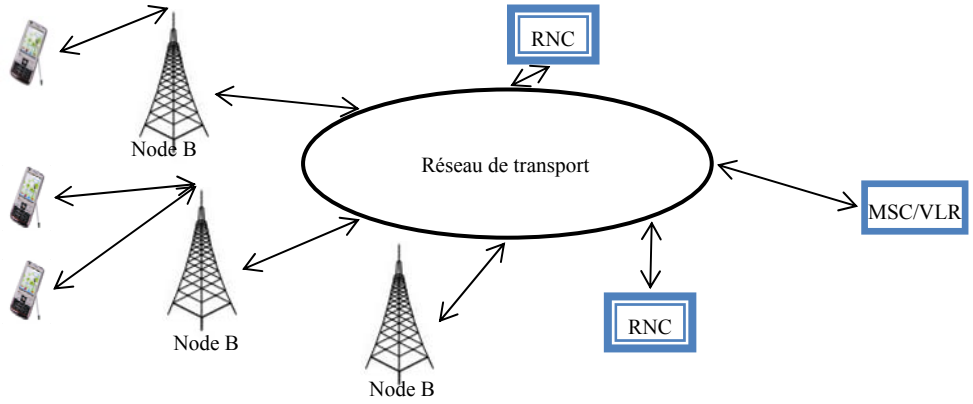
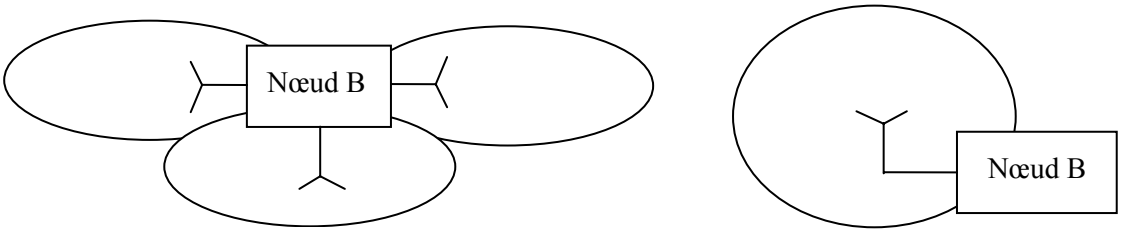


Figure B.3: Architecture générale de l'UTRAN

Nœud B

Le nœud B assure toutes les fonctions de la couche physique sur la voie radio: transmission et réception, modulation, démodulation, étalement de spectre, codage correcteur, etc. Il prend également en charge le contrôle de puissance du mobile. Comme dans le GSM, la réalisation technique et l'architecture interne du Nœud B sont laissées à l'appréciation du constructeur. Il est possible d'utiliser la tri-sectorisation ou d'utiliser des antennes omnidirectionnelles, voir figure B.4.



Nœud B avec tri-sectorisation

Nœud B avec antenne omnidirectionnelle

Figure B.4 : Implémentations possibles du nœud B

La norme a repris le vocabulaire courant et considère qu'il y a un seul nœud B qui couvre plusieurs secteurs dans le cas de la tri-sectorisation.

Le RNC

Le RNC de l'UTRAN a une fonction équivalente au BSC des réseaux GSM. Il assure principalement le routage des communications entre le Nœud B et le réseau cœur. Il contrôle l'utilisation et l'intégrité des ressources radio du Nœud B en question. L'existence d'un réseau de transport, permet en théorie, à un Nœud B d'avoir des liaisons avec plusieurs RNC. Cependant, en pratique, cela n'est pas le cas, car ceci n'est pas prévu par les recommandations. Un nœud B est contrôlé par un seul RNC appelé Controlling RNC. Par la suite et afin de simplifier le vocabulaire quand nous décrivons les actions effectuées par un RNC sur un nœud B, nous utilisons le terme RNC pour designer le Controlling RNC.

Les actions effectuées par un RNC pour un mobile donné, d'après le TS 25.401, sont:

- Etablissement d'une connexion de contrôle, RRC (Radio Resource Control) entre le mobile et le RNC.
- Affectation des ressources radio.
- Gestion du contrôle de la puissance.
- Gestion de la configuration ou reconfiguration de l'interface radio et de la mobilité du mobile (handover) qui nécessite un échange de signalisation.

Les spécifications distinguent différents types de RNC en fonction de leur rôle respectif pour chaque communication:

- Serving RNC: est le RNC qui a une connexion RRC avec le terminal. Il est celui qui effectue la gestion des connexions radio, le raccordement avec le réseau cœur et qui contrôle et exécute le handover.
- Drift RNC: est un RNC qui assure la liaison entre le mobile et le serving RNC. Il joue le rôle de simple routeur vis-à-vis des données RRC.
- Controlling RNC: est le RNC qui contrôle le Nœud B responsable de la cellule ou se retrouve le mobile. Tous ces types de RNC sont illustrés dans la figure B.5.

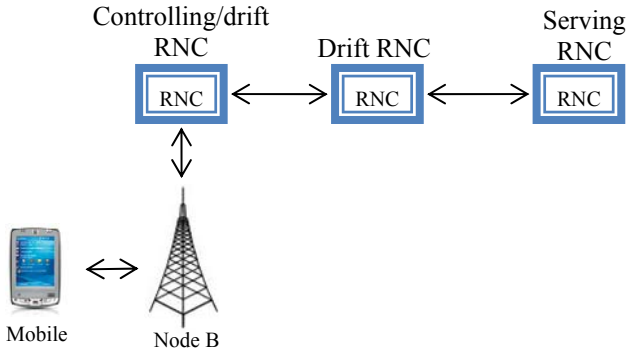


Figure B.5: Rôles du RNC

B.2.3. Interfaces du réseau d'accès

Les interfaces du réseau d'accès sont désignées par les lettres Iu:

- Iu-CS: est l'interface entre le RNC et le domaine CS du réseau cœur.
- Iu-PS: est l'interface entre le RNC et le domaine PS du réseau cœur.
- Iub: est l'interface entre le Nœud B et le RNC.
- Iur: est l'interface entre deux RNC. Elle est la seule qui n'a pas d'équivalent en GSM.

Chacune des ces interfaces supporte deux types des protocoles: AP (Application Protocol), pour les échanges de signalisation entre les équipements et le FP (Frame Protocol) pour les données usager.

Il y a aussi une interface Uu qui relie le mobile et le réseau UTRAN. Elle assure la liaison entre le mobile et le RNC. Le Nœud B est utilisé seulement pour transporter les données.

Ces différentes interfaces sont représentées dans la figure B.6.

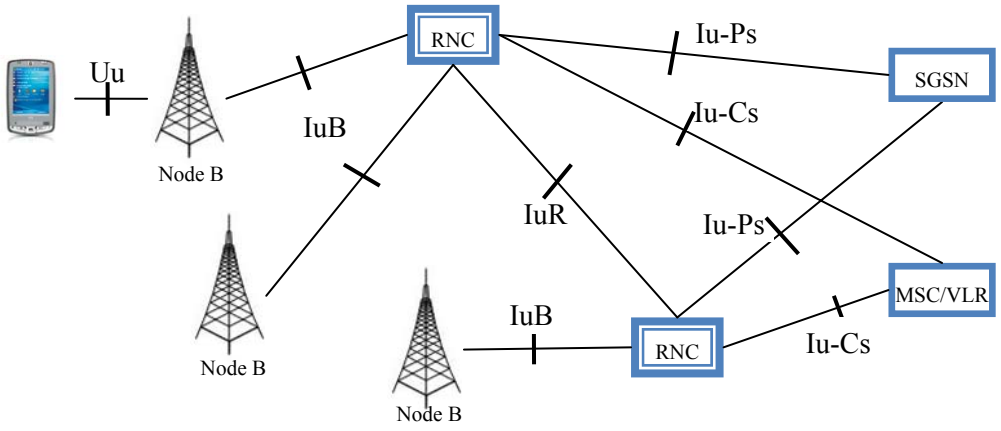


Figure B.6: Interfaces dans le réseau d'accès UTRAN

Un des protocoles le plus important pour la signalisation est le protocole RANAP (Radio Access Network Application Part). Il est utilisé sur l'interface Iu-PS ou sur l'interface Iu-CS pour envoyer de messages de signalisation entre le réseau d'accès et le réseau cœur.

Un autre type de connexion importante est le RRC. C'est une connexion entre l'UE et le réseau d'accès transportant des messages de contrôle et elle est situé au couche réseau selon le model OSI. Elle inclue les fonctions telles que l'établissement et la terminaison des connexions, la diffusion d'information du système ou l'établissement et la reconfiguration des caractéristiques de la porteuse radio.

B.3. Réseau cœur

Dans l'architecture GSM, le réseau cœur était scindé en deux parties distinctes correspondant à un découpage entre les services de commutation de circuits et ceux réservés aux commutations des paquets. La conséquence de cette approche est la gestion séparée de l'établissement d'appel, de la mobilité de l'abonné et de la sécurité pour chaque partie.

Cette séparation subsiste dans la norme UMTS. Ainsi, les spécifications UMTS parlent de «domaines» de service. La version 99 des spécifications UMTS définissent deux domaines: le domaine CS (Circuit Switched domain) et le domaine PS (Packet Switched domain).

B.3.1 Domaines du réseau

Les éléments du réseau cœur sont classifiés en trois groupes. Le première est celui des éléments du domaine CS et comprend le MSC, le GMSC (Gateway MSC) et le VLR (Visitor Location Register). Le deuxième est le domaine PS qui comprend le SGSN (Serving GPRS Support Node) et le GGSN (Gateway GPRS Support Node). Le dernier groupe comprend les éléments du réseau communs au deux domaines: le HLR (Home Location Register), l'EIR (Equipment Identity Register) et l'AuC (Authentication Center). Cette structure est montrée dans la figure B.7.

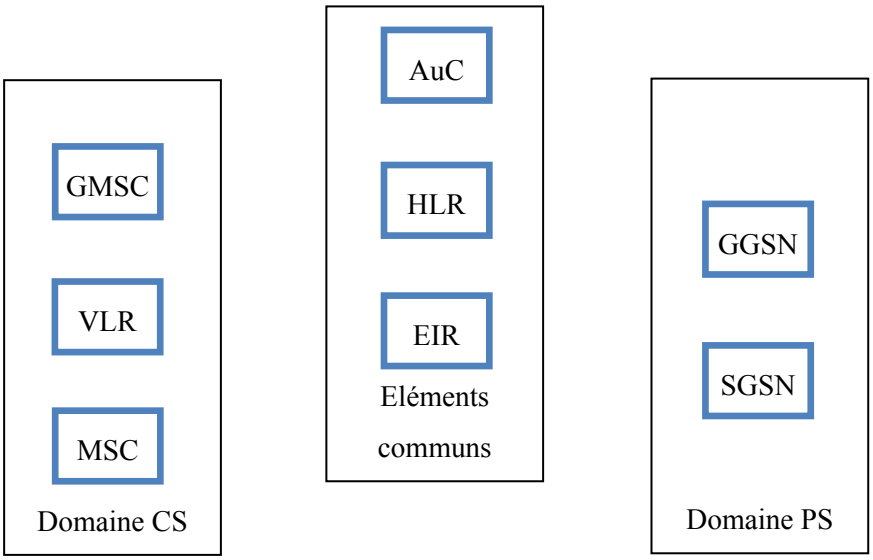


Figure B.7: Domaines du réseau cœur

Dans le domaine CS, le commutateur MSC prend en charge un certain nombre des cellules par l'intermédiaire des contrôleurs de stations de base (RNC) de l'UTRAN. Il s'occupe d'une zone et des mobiles qui s'y trouvent grâce à une base des données locale le VLR. Le GMSC est un MSC qui assure une fonction de passerelle. Il s'occupe de l'interconnexion du réseau UMTS avec un réseau de téléphonie fixe PSTN (Public Switched Telephone Network) ou un autre réseau mobile. En pratique la fonction de passerelle se trouve intégrée dans tous les MSC du réseau.

Les deux entités du domaine PS sont les GSN (GPRS Support Node):

- Serveur (SGSN).
- Passerelle (GGSN).

Les deux sont des routeurs IP, enrichis de fonctions spécifiques et d'interfaces leur permettant de communiquer avec les autres entités du réseau. En particulier, le SGSN doit être capable d'échanger les informations avec le HLR et doit, par conséquent, disposer des couches protocolaires utilisées pour la signalisation dans le réseau cœur.

Le SGSN est l'équivalent du MSC/VLR du domaine CS. Il gère un ensemble des zones de routage et s'interconnecte au sous-système radio, par l'intermédiaire des contrôleurs de station de base (RNC). Il est responsable de la gestion des terminaux qui se trouvent dans une des zones de routages qu'il gère et s'occupe de la procédure d'attachement, de l'établissement d'un contexte PDP (Packet Data Protocol) et de la collecte des informations de taxation. C'est aussi le SGSN qui est en charge de la gestion de la mobilité et de la compression des données échangées avec les terminaux.

Le GGSN est la passerelle entre le domaine PS et un autre réseau UMTS ou un réseau des données PDN (Packet Data Network). Les données échangées entre le mobile et le PDN passent par le GGSN.

Le domaine des éléments communs est constitué des bases de données indispensables au fonctionnement d'un réseau UMTS. En premier lieu, le HLR stocke les informations sur les abonnements et sur la localisation d'une carte USIM, c'est-à-dire l'identité du MSC où se trouve le mobile contenant la carte. Le serveur d'authentification AuC fournit les clés d'authentification et est associé à un HLR. La base de données EIR contient la liste de terminaux autorisés à fonctionner dans le réseau (liste blanche) et celle des terminaux interdits (liste noire). Les terminaux sont identifiés dans l'EIR par une identité unique attribuée par le fabricant : l'IMEI (International Mobile Equipment Identity). La vérification de l'IMEI et la consultation de l'EIR ne sont pas imposées par les recommandations.

B.3.2. Réseau d'origine et réseau de service

Le réseau d'origine est composé des éléments HLR, AuC et EIR. Il appartient toujours au fournisseur des services UMTS, auprès duquel l'abonné est inscrit et auquel il fait le plus de confiance.

Le réseau de service est composé des autres éléments du réseau cœur : MSC, VLR, GMSC, SGSN et GGSN. Si l'utilisateur est en itinérance les éléments du réseau de service vont appartenir au réseau qui accueille le mobile et auquel le mobile fait le moins de confiance.

Du point de vue de la sécurité, la division en domaines (CS et PS) n'est pas très importante car toutes les procédures de sécurité se déroule de manière indépendantes, mais similaires pour les deux domaines. La division réseau d'origine/réseau de service est beaucoup plus importante car le niveau de confiance de l'utilisateur dans les deux domaines est différent.

Références

Références

Al-Saraireh, J. Yousef, S. Al Nabhan, M., 2006. Enhancement Mobile Security and User Confidentiality for UMTS. *Second European Conference on Mobile Governement*, Brighton, Grande Bretagne.

Alvarez, G. Li, S., 2006. Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems, *International Journal of Bifurcation and Chaos*, Volume 16, Issue 8, pp 2129-2151.

Awad, A. El Assad, E. Caragata, D. Bakhache, B., 2007. *Rapport d'étude sur quelques methodes de chiffrage/déchiffrage bassées chaos*, Raport scientifique, Projet ANR ACSCOM.

Awad, A. El Assad, E. Caragata, D. Noura, H., 2008. *Etude comparative de deux algorithms de chiffrement chaotique vis-à-vis de la cryptanalyse et des erreurs de propagation*, Raport scientifique, Projet ANR ACSCOM.

Awad, A. Ahmad, K. El Assad, S. Caragata, D., 2010. Chaos-Based Cryptosystem for Secure Transmitted Images, *International Conference on Telecommunications and Multimedia (TEMU)*, Crète, Grèce.

Bais, A. Penzhorn, W. T. Palencky, P., 2006. Evaluation of UMTS security architecture and services. *IEEE International Conference on Industrial Informatics*, Singapore.

Barkan, E. Biham, E. Keller, N., 2003. Instant ciphertext-only cryptanalysis of GSM encrypted communications. *Advances in Cryptology, Proceedings of Crypto 2003*, pp. 600-616.

Barbu, I. Sofron, E., 2009. Code Division Multiple Access system based on Chaos Spreading Sequence. *International Conference on Electronics, Computers and Artificial Intelligence* (ECAI'2009), Pitesti, Roumanie.

Bem, J.M. Wieckowski, T.W. Zielinski R.J., 2000. Broadband Satellite Systems. *IEEE Communications Survey and Tutorials*, volume 3, pp 2-15.

Billet, O. Gilbert, H., 2005. Resistance of SNOW 2.0 against Algebraic Attacks. *Lecture Notes in Computer Science, The Cryptographer's Track at RSA Conference*, Springer, pp 19-28.

Bouchard, G., 2007. Directives pour la mise en réseau des trames de transport. *Revue des technologies de Radio-Canada*, numero 3, pp 1-12.

Bremaud, P., 2008. *Markov Chains*, Springer, Corrected Edition (1 may 2008).

Caragata, D. El Assad, S. Serbanescu, A., 2006. *Development of some encryption/decryption algorithms using chaos for secure communication systems*. Internal Research Report of IREENA Laboratory/ Projet Licence Académie Technique Militaire.

Caragata, D. Bakhache, B. El Assad, S. Tutanescu, I., 2009a. Security Enhancement for Internet Communications over Satellite DVB using Chaos, *Lecture Notes in Engineering and Computer*

Science: Proceedings of The World Congress on Engineering and Computer Science, San Francisco, Etats-Unis, pp. 419-424.

Caragata, D. El Assad, S. Tutanescu, I, Sofron, E., 2009b. Secure TCP/IP Communications over DVB-S/DVB-RCS Using Chaotic Sequences, *The 4th International Conference for Internet Technology and Secured Transactions*, Londres, Grande Bretagne.

Caragata, D. El Assad, S. Bakhache, B. Tutanescu, I., 2010a. Secure IP over Satellite DVB Using Chaotic Sequences. *Engineering Letters*, Vol. 18, Issue 2, pp. 135-146.

Caragata, D. El Assad, S. Noura, H. Tutanescu, I., 2010b. Secure Unicast and Multicast over Satellite DVB Using Chaotic Generators, *International Journal for Internet Technologies and Secure Transactions*, Inderscience Publishers, vol. 2, number 3-4, pp 357-379.

Caragata, D. El Assad, S. Tutanescu, I. Sofron, E., 2010c. Chaos Based Secure IP Communications over Satellite DVB, *IAENG Transactions on Engineering Technologies Volume 4 - Special Edition of the World Congress on Engineering and Computer Science 2009*, American Institute of Physics, pp 26-40.

Caragata, D. Sofron, E. Tutanescu, I. El Assad, S., 2010d. Secure IP Multicast over Satellite. *The Third International Symposium on Electrical and Electronics Engineering (ISEEE-2010)*, Galati, Roumanie.

Caragata, D. Radu A.L. El Assad, S. Apostol, C., 2010e. Chaos Based Fragile Watermarking Algorithm for JPEG Images, *The 5th International Conference for Internet Technology and Secured Transactions (ICITST-2010),* Londres, Grande Bretagne.

Caragata, D. Tutanescu, I. El Assad, S. Shoniregun C. A., 2011. Security of Mobile Internet Access with UMTS/HSDPA/LTE, *World Congress on Internet Security*, Londres, Grande Bretagne.

CDMA Development Group, 2010. *CDMA Deployment Search.* [Online] Disponible: http://www.cdg.org/worldwide/index.asp?h_area=0&h_technology=999&h_frequency=1

Clark A.C., 1945. Extra-Terrestrial Relays - Can Rocket Stations Give Worldwide Radio Coverage. *Wireless World*.

Collini-Nocker, B. Fairhurst, G., 2004. ULE versus MPE as an IP over DVB Encapsulation. *Performance Modelling and Evaluation of Heterogeneous Networks*, West Yorkshire, Grande Bretagne.

Conseil de L'Union Européenne, 1995. *Résolution du conseil du 17 janvier 1995 relative à l'interception légale des télécommunications (96/C 329/01).*

Cruickshank, H. Howarth, M.P. Iyengar, S. Sun, Z., 2005. A comparison between satellite DVB Conditional Access and secure IP multicast. *14th IST Mobile and Wireless Communications Summit*, Dresden, Allemagne.

DIX (Digital Equipment Corp, Intel Corp, Xerox Corp), 1982, "Ethernet Local Area Network Specification" Version 2.0.

Debraize, B. Corbella, I. M., 2009. Fault Analysis of the Stream Cipher Snow 3G. *Workshop on Fault Diagnosis and Tolerance in Cryptography*, Lausanne, Switzerland.

Diffie, W. Hellman, M. E., 1976. New Directions in Cryptography, *IEEE Transactions on Information Theory*, Vol. 22, Issue 6, pp. 644-654.

ECRYPT II Network of Excellence, 2010. *Yearly Report on Algorithms and Keysizes (2009-2010)*.

El Assad, S. Noura, H. Taralova, I., 2008. Design and Analyses of efficient chaotic generators for crypto-systems, *IAENG Special Edition of the World Congress on Engineering and Computer Science 2008, WCECS '08. Advances in Electrical and Electronics Engineering*, Vol. 1, pp 3-12.

El Assad, S. Noura, H. Caragata, D., 2010a. A Fast and Robust Chaos-Based Cryptosystem for Transmitted Data. *The 11th Experimental Chaos and Complexity Conference*, Lille, France.

European Telecommunications Standards Institute (ETSI), 1997. *EN 300 421, Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services*.

European Telecommunications Standards Institute (ETSI), 2004. *EN 301 192 – Digital Video Broadcasting (DVB); DVB specification for data broadcasting*.

European Telecommunications Standards Institute (ETSI), 2005. *EN 302 307, Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications*.

European Telecommunications Standards Institute (ETSI), 2005. *EN 301 790, Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems*.

European Telecommunications Standards Institute (ETSI), 2009. *TS 102.310 Smart Cards; Extensible Authentication Protocol support in the UICC.*

Fairhurst, G. Matthewus, A., 2003. A comparison of IP transmission using MPE and a new Lightweight Encapsulation. *IEEE Seminar on IP Over Satellite - The next Generation: MPLS, VPN and DRM Delivered Services*, pp.106-120.

Ferguson, N. Schneider B., 2004. *Cryptographie en pratique*. Vuibert.

Fumy, W. Landrock, P., 1993. Principles of key management. *IEEE Journal on selected areas in communications*. Vol. 11, no. 5.

Géron, A., 2006. *WIFI, Déploiement et sécurité.* Dunod, pp. 303-340.

Hong, T.C. Chee, W.C. Budiarto, R., 2005. A comparison of IP Datagrams Transmission using MPE and ULE over MPEG/DVB Networks. *Fifth International Conference on Information, Communications and Signal Processing*, Bangkok, Thaïlande.

Horn, G. Martin, K.M. Mitchell, C., 2002. Authentication protocols for mobile network environment value-added services. *IEEE Transactions on Vehicular Technology*, vol. 51, no. 2, pp. 383-392.

Howarth, M. P. Iyengar, S. Sun, Z. Cruickshank, H., 2004. Dynamics of Key Management in Secure Satellite Multicast. *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 2, pp. 308-319.

Hu Kwong, Y. Jiwu Huang, S., 2004. Using invisible watermarks to protect visibly watermarked images, *IEEE International Symposium on Circuits and Systems, ISCAS'04*, Vancouver, Canada.

Huang, J. Shi, Y.Q. Shi, Y., 2000. Embedding Image Watermarks in DC Components, *IEEE Transactions on Circuits and Systems for Video Technology*, Volume 10, pp 974-979.

Hubenko, V. P. Raines, A. R. Baldwin, R. O. Mullins, B. E. Mills, R. F. Grimaila, M. R., 2007. Improving Satellite Multicast Security Scalability by Reducing Rekeying Requirements. *IEEE Network Magazine*, volume 21, issue 4, pp 51-56.

Internet Engineering Task Force (IETF), 2000. *RFC 2994, A Description of the MISTY1 Encryption Algorithm.* Ohta, H., Matsui, M.

Internet Engineering Task Force (IETF), 2008. *Security Extension for Unidirectional Lightweight Encapsulation Protocol, Internet Draft.* Cruickshank, H. Pillai, P. Iyengar, S.

Internet Engineering Task Force (IETF), 2009. *RFC 5458, Security Requirements for the Unidirectional Lightweight Encapsulation (ULE) Protocol.* Cruickshank H., Pillai P., Noisternig M., Iyengar S.

Internet Engineering Task Force (IETF), 2005. *RFC 4326, Unidirectional Lightweight Encapsulation (ULE) for Transmission of IP Datagrams over an MPEG-2 Transport Stream (TS).* Fairhurst G., Collini-Nocker B.

Internet Engineering Task Force (IETF), 2005. *RFC 4301, Security Architecture for the Internet Protocol*, Kent S., Seo K.

Internet Engineering Task Force (IETF), 2005. *RFC 4302, IP Authentication Header*, Kent S.

Internet Engineering Task Force (IETF), 2005. *RFC 4303, IP Encapsulating Security Payload (ESP).* Kent S.

Internet Engineering Task Force (IETF), 2005. *RFC 4306, Internet Key Exchange (IKEv2) Protocol.* Kaufman C.

Internet Engineering Task Force (IETF), 2005. *RFC 4259, A Framework for Transmission of IP Datagrams over MPEG-2 Networks.* Montpetit, M.J. Fairhurst, G. Clausen, H. Collini-Nocker, B. Linder, H.

Internet Engineering Task Force (IETF), 1998. *RFC 2412, The OAKLEY Key Determination Protocol.* Orman H.

Internet Engineering Task Force (IETF), 1998. *RFC 2401, Security Architecture for the Internet Protocol.* Kent S.

Iyengar, S. Cruickshank, H. Pillai, P. Fairhurst, G. Duquerroy, L., 2007. Security requirements for IP over satellite DVB networks. *16th IST Mobile and wireless Communications Summit*, Budapest, Hongrie.

Kambourakis, G., 2005. Issues on 2/2.5/3 (3GPP) Mobile Networks Security. *Intensive Programme on Information and Communication Systems Security*, University of the Aegean, Laboratory of Information & Communications Systems Security, Greece.

Katzenbeisser, S. Petitcolas, F. A., 2000. *Information Hiding techniques for steganography and digital watermarking*, Artech House.

Kerckhoffs, A., 1883. La cryptographie militaire. *Journal des sciences militaires*, vol. IX, pp. 5–38.

Khan, M. Ahmed, A. Cheema, A. R., 2008. Vulnerabilies of UMTS Access Domain Security Architecture. *Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, Thailand.

Li, S. Chen, G. Mou, X., 2005. On the Dynamical Degradation of Digital Piecewise Linear Chaotic Maps, *International Journal of Bifurcation and Chaos*, vol. 15, no. 10, pp. 3119-3151.

Menezes, A. van Oorschot, P. Vanstone, S., 1996. *Handbook of Applied Cryptographie*. CRC Press.

Meyer, U. Wetzel, S., 2004a. A Man-in-the-middle attack on UMTS. *Proceedings of 5th International Conference on Web Information System Engineering*, Australia.

Meyer, U. Wetzel, S., 2004b. On the impact of GSM encryption and man-in-the-middle attacks on the security of interoperating GSM/UMTS networks. *Personal, Indoor and Mobile Radio Communications*, vol. 4, pp. 2876-2883.

Mittra, S., 1997. Iolus: A framework for scalable secure multicasting. *Proceedings of SIGCOM*, pp. 277-288, Cannes, France.

Mohanty, S. P., 1999. *Digital Watermarking: A tutorial review*, Rapport technique de Department of Computer Science and Engineering of the University of South Florida, Etats-Unis.

Nafornita, C., 2008. *Contributii la marcarea transparenta a imaginilor in domeniul transformatei wavelet*, Thèse de doctorat, Universitatea Tehnica Cluj-Napoca et Universitatea Politehnica Timisoara.

National Institute of Standards and Technology (NIST), 2001a. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. NIST Special Publication 800-22.

National Institute of Standards and Technology (NIST), 2001b. *Announcing the Advanced Encryption Standard (AES)*. Federal Information Processing Standards Publication 197.

National Institute of Standards and Technology (NIST), 2006. *Recommendation for Key Management – Part 1: General (Revised). NIST Special Publication.* Barker, E. Barker, W. Burr, W. Polk, W. Smid, M.

Noisterling, N. Collini-Nocker, B., 2007. A lightweight security extension for ULE. *International Workshop on Satellite and Space Communications*, Salzburg, Autriche.

Noura, H. Henaff, S. Taralova, I. El Assad, S., 2009. Efficient cascaded 1-D and 2-D chaotic generators, *Second IFAC Conference on Analysis and Control of Chaotic Systems Chaos*, Londres, Grande Bretagne, 22-24 juin 2009.

Noura, H., thèse en cours. "Crypto compression basé chaos pour la sécurité des communications mobiles".

Pillai P. Hu, Y.F., 2006. Design and Analysis of Secure Transmission of IP over DVB-S/RCS Satellite Systems. *IFIP International Conference on Wireless and Optical Communications Networks*, Bangalore, Inde.

Rincu, C.I. Serbanescu, A. Mateescu, A., 2006. Chaos based cryptography. Past and trends. *Proceedings of Communications*, Bucarest, Roumanie.

Rincu, C.I., 2007a. *Contributii la protectia informatiei in retelele de comunicatii. Tehnici bazate pe sistemele dinamice haotice*. Thèse de doctorat, Académie Technique Militaire.

Rincu, C.I. Iana, V.G. Serban, G. Tutanescu, I., 2007b. A Chaotic Generator Implemented in Reprogrammable Hardware. *Applied Electronics International Conference*, Pilsen, Allemagne.

Rincu, C.I., 2009. *Aplicatii ale sistemelor dinamice haotice in criptografie*. Editura Academiei Tehnice Militare, Bucuresti.

Sattarzadeh, B. Asadpour, M. Jalili, R., 2007. Improved User Identity Confidentiality for UMTS Mobile Networks. *Fourth European Conference on Universal Multiservice Networks (ECUMN'07)*, Toulouse, France.

Schneier, B., 2001. *Applied Cryptography*. Second Edition, John Wiley & Sons, Inc.

Secrétariat général de la défense nationale, 2006. *Mécanismes cryptographiques – Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques de niveau de robustesse standard*, Direction centrale de la sécurité des systèmes d'information, Paris.

Serbanescu, A., 2004. *Aplicatii ale sistemelor dinamice haotice in comunicatii*. L'Académie Technique Militaire, Roumanie.

Serbanescu, A. Rincu, C.I., 2008. *Systèmes et signaux face au chaos. Applications aux communications.* L'Académie Technique Militaire, Roumanie.

Shirai, T. Mizuno, A., 2008. A Compact and High Speed Cipher Suitable for Limited Resource Environment. *3rd ETSI Security Workshop*, Sophia Antipolis, France.

Smart, N., 2009, *D.SPA.7 Yearly Report on Algorithms and Keysizes (2009)*. Raport technique du projet : ECRYPT II.

Socek, D. Li, S. Magliveras, S.S. Furht, B., 2005, Enhanced 1-D Chaotic Key-Based Algorithm for Image Encryption. *Proceedings of the First IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm 2005),* pp. 406-408, Athènes, Grèce.

Sun, Z. Howarth, M.P. Cruickshank, H. Iyengar, S., 2003, Networking Issues in IP Multicast over Satellite, *International journal of Satellite Communications and Networking*, Vol. 21, No. 4-5, pp. 489-507.

Tefas, A. Nikolaidis, A. Nikolaidis, N. Solachidis, V. Tsekeridou, S. Pitas, I., 1999. Performance Analysis of Watermarking Scheme Based on Skew Tent Chaotic Sequences, *European Project IST CERTIMARK*, 1999.

Thomos, N. Boulgouris, N. V. Strintzis, M. G., 2006. Optimized Transmission of JPEG2000 Streams Over Wireless Channels, *IEEE Transactions on Image Processing*, vol 15, pp 54-67.

Tsekeridou, S. Nikolaidis, N., 2000. Copyright protection of still images using self-similar chaotic watermarks, *Proceedings of IEEE International Conference on Image Processing (ICIP'00), Volume 1, pp 411-414*, Vancouver, Canada.

Tutanescu, I. Anton, C. Caragata D., 20110. Use of Elliptic Curves Cryptosystems in Information Security, *5th International Conference on Information Technology, ICIT'11*, Al-Zaytoonah University, Jordanie.

Waldvogel, M. Caronni, G. Sun, D. Weiler, N. Plattner, B., 1999, The VersaKey framework: Versatile Group Key Management. *Journal on Selected Areas in Communications: Special Issue on Middleware*, vol. 19, no. 9, pp. 1614-1634.

Wang, H. Ding, K. Liao, C., 2008. Chaotic watermarking scheme for authentication of JPEG Images, *International Symposium on Biometrics and Security Technologies*, Islamabad, Pakistan, 23-24 avril 2008.

Watanabe, D. Biryukov, A. De Cannière, C., 2004. A distinguishing Attack of SNOW 2.0 with Linear Masking Method. *Lecture Notes in Computer Science, Selected Areas in Cryptography*, Springer, pp. 222 – 233.Wenyuan Xu Trappe, W. Paul S., 2004. Key management for 3G

MBMS security. *IEEE Global Telecommunications Conference, (GLOBECOM)*. Vol. 4, Issue 29 pp 2276 – 2280.

Wong, C.W. Gouda, M. Lam, S.S., 2000. Secure Group Communications Using Key Graphs. *IEEE/ACM Transactions on Networking*, vol. 8, pp. 16-30.

Yantao, Z. Yunfei, M. Zhiquan, L., 2008. A robust chaos-based DCT domain watermarking algorithm, *Proceedings of IEEE International Conference on Computer Science and Software Engineering, Volume 3, pp 935-938*, Wuhan, China.

3rd Generation Partnership Project, 2001. *TS 21.133; Security threats and requirements.*

3rd Generation Partnership Project, 2001. *TS 33.120; Security principles and objectives.*

3rd Generation Partnership Project, 2010. *TS 22.022; Personalisation of Mobile Equipment (ME); Mobile functionality specification.*

3rd Generation Partnership Project, 2003. *TS 22.048; Security mechanisms for the (U)SIM Application Toolkit; Stage 1.*

3rd Generation Partnership Project, 2005. *TS 23.048; Security mechanisms for the (U)SIM application toolkit; Stage 2.*

3rd Generation Partnership Project, 2010. *TS 23.060; General Packet Radio Service (GPRS);*

Service description.

3rd Generation Partnership Project, 2010. *TS 25.331; Radio Resource Control (RRC); Protocol Specification.*

3rd Generation Partnership Project, 2010. *TS 25.401; UTRAN overall description.*

3rd Generation Partnership Project, 2010. *TS 27.007; AT command set for User Equipment (UE).*

3rd Generation Partnership Project, 2010. *TS 29.002; Mobile Application Part (MAP) specification.*

3rd Generation Partnership Project, 2009. *TS 31.101; UICC-terminal interface; Physical and logical characteristics.*

3rd Generation Partnership Project, 2010. *TS 31.103; Characteristics of the IP Multimedia Services Identity Module (ISIM) application.*

3rd Generation Partnership Project, 2009. *TS 31.111; Universal Subscriber Identity Module (USIM) Application Toolkit (USAT)*

3rd Generation Partnership Project, 2009. *TS 33.105; Cryptographic algorithm requirements*

3rd Generation Partnership Project, 2001. *TS 33.103; 3G security; Integration guidelines.*

3rd Generation Partnership Project, 2007. *TS 33.200; 3G Security; Network Domain Security; MAP application layer security.*

3rd Generation Partnership Project, 2009. *TS 35.205 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1: General.*

3rd Generation Partnership Project, 2009. *TS 35.206; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm Specification.*

3rd Generation Partnership Project, 2009. *TS 35.207; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 3: Implementors' Test Data.*

3rd Generation Partnership Project, 2009. *TS 35.208; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 4: Design Conformance Test Data.*

3rd Generation Partnership Project, 2009. *TS 35.215; Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2; Document 1: UEA2 and UIA2 specifications.*

3rd Generation Partnership Project, 2009. *TS 35.216; Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2; Document 2: SNOW 3G specification.*

3rd Generation Partnership Project, 2009. *TS 35.217; Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2; Document 3: Implementors' test data.*

3rd Generation Partnership Project, 2009. *TS 35.218; Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2; Document 4: Design conformance test data.*

3rd Generation Partnership Project, 2001. *TR 33.908; 3G Security; General Report on the Design, Speification and Evaluation of 3GPP Standard Confidentiality and Integrity Algorithms.*

3rd Generation Partnership Project, 2000. *TR 33.900; A Guide to 3rd Generation Security.*

Liste des abréviations

Liste des abréviations

3GPP – 3rd Generation Partnership Project

AA – Authentication Algorithm

AAL5 – ATM Adaptation Layer 5

AC – Alternative Coefficient

ACM – Adaptive Coding and Modulation

ADSL – Assymetric Digital Subscriber Line

AES – Advanced Encryption Standard

AH – Authentication Header

AI – Added Information

AK – Anonymity Key

AKA – Authentication and Key Agreement

AMF – Authentication and key Management Field

AP – Aplication Protocol

ARIB – Association of Radio Radio Industries and Business

ATIS – Alliance for Telecommunications Industry Solutions

ATM – Asynchronous Transfer Mode

ATSC – Advanced Telecommunications System Committee

AuC – Authentication Center

AUTN – Authentication token

AV – Authentication Vector

BSC – Base Station Controller

BSS – Base Station Subsystem

BTS – Base Transciever Station

CBC – Cipher Block Chaining

CCSA – China Communications Standards Association

CIA – Central Intelligence Agency

CK – Cipher Key

CM – Commercial Module

CompuSec – Computer Security

ComSec – Communications Security

CRC – Cyclic Redundency Check

CryptoSec – Cryptographic Security

CS – Circuit Switched

DC – Direct Coefficient

DCT – Dicrete Cosine Transform

DES – Data Encryption Standard

DF – Dedicated File

DFRCBCSIM – Design of a Fast and Robust Chaos Based Crypto-System for Image Encryption

DO – Data Overhead

DOD – Department of Defence

DoS – Denial of Service

DVB – Digital Video Broadcasting

DVB-C – DVB Cable

DVB-H – DVB Hanhelds

DVB-RCS – DVB Return Channel via Satellite

DVB-S – DVB Satellite

DVB-T – DVB Terrestrial

EA – Encryption Algorithm

EAP – Extension Authentication Protocol

ECKBA – Enhanced 1-D Chaotic Key-Based Algorithm

EDGE – Enhanced Data rates for GSM Evolution

EEPROM – Electronicaly Erasable Programable Read Only Memory)

EF – Elementary File

EHI – Extension Header Information

EIR – Equipment Identity Register

EK – Ephemeral Key

EmSec – Emission Security

ES – Elementary Stream

ESP – Encapsulating Security Payload

ETSI – European Telecommunications Standard Institute

FEC – Forward Error Correction

FIFO – First Input First Output

FP – Frame Protocol

FSM – Finite State Machine

FTP – File Transfer Protocol

GC – Group Controller

GF – Galois Field

GGSN – Gateway GPRS Support Node

GM – Group Member

GMSC – Gateway MSC

GPRS – General Packet Radio Service

GSE – Generic Stream Encapsulation

GSM – Global System for Mobile communications

GSN – GPRS Support Node

HF – Hash Function

HLR – Home Location Register

HSDPA – High Speed Downlink Packet Access

HSUPA – High Speed Uplink Packet Access

IANA – Internet Assigned Numbers Authority

IETF – Internet Engineering Task Force

IIR – Infinite Impulse Response

IK – Integrity Key

IKE – Internet Key Exchange

IKi – Intermediary Key level 1

IKiTH - Intermediary Key level 1 Threshold

IMEI – International Mobile Equipment Identification

IMSI – International Mobile Subscriber Identity

InfoSec – Information Security

IP – Internet Protocol

IPPRU – IP Packet Recovery Unit

IPSec – IP Security

IS-95 – Interim Standard 95

ISAKMP – Internet Security Association Key Management Protocol

ISDB – Integrated Services Digital Broadcasting

ISIM – IP Multimedia Services Identity Module

ISP – Internet Service Provider

ITV – International TV

JPEG – Joint Photographic Experts Group

KAC – Key Administrator Center

KDC – Key Distribution Center

KE – Key Exchange

KEK – Key Encryption Key

KL – Key Level

KMI – Key Management Information

KSI – Key Set Identifier

KTC – Key Translation Center

LAI – Local Area Identification

LAN – Local Area Network

LFSR – Linear Feedback Shift Register

LKH – Logical Key Hierarchy

LSB – Least Significant Bit

LTE – Long Term Evolution

MAC – Media Access Control

MAC – Message Authentication Code

MAP – Mobile Application Part

MAPSec – MAP Security

ME – Mobile Equipment

MF – Master File

MF-TDMA – Multi Frequency – Time Division Multiple Access

MK – Master Key

MPDU – Multicast PDU

MPE – Multi Protocol Encapsulation

MPEG – Moving Pictures Experts Group

MS – Mobile Station

MSC – Mobile Switching Center

MT – Mobile Termination

NetSec – Network Security

NIST – National Institute for Standards and Technology

NPA – Network Point of Attachment

OU – Ocasional Use

PAT – Program Allocation Table

pcap – packet capture format

PDC – Personal Digital Cellular

PDN – Packet Data Network

PDP – Packet Data Protocol

PDU – Packet Data Unit

PID – Packet Identifier

PIN – Personal Identification Number

PMK – Period of MK usage

PMT – Program Map Table

PN – Packet Number

PP – Payload Pointer

PS – Packet Switched

PSI – Program Specific Information

PSNR – Peak Signal to Noise Ratio

PSTN – Public Switched Telephone Network

PUSI – Packet Unit Start Indicator

PWK – Pair Wise Key

PWLCM – Piece Wise Linear Chaotic Map

RAI – Routing Area Identification

RAM – Random Access Memory

RANAP – Radio Access Network Application Part

RAND – Random number

RES – Response

RGB – Red Green Blue

RNC – Radio Network Controller

ROM – Read Only Memory

RRC – Radio Resource Control

SA – Security Association

SAD – Security Association Database

SAGE – Security Algorithm Experts Group

SGSN – Server GPRS Support Node

SK – Session Key

SKEME – RFC – Request for Comments

SKTh – Session Key Threshold

SMS – Short Message Service

SN – Sequence Number

SNDU – Sub Network Data Unit

SPD – Security Policy Database

SPDU – Security PDU

SPI – Security Parameter Index

SQN – Sequence Number

SS7 – Signalling System no. 7

TE – Terminal Equipment

TEK – Traffic Encryption Key

TI – Total Information

TK – Temporary K

TLS – Transport Layer Security

TM – Technical Module

TMSI – Temporary Mobile Subscriber Identity

TranSec – Transmission Security

TS – Transport Stream

TTA – Telecommunications Technology Association

TTC – Telecommunications Technology Committee

UE – User Equipment

UEA – UMTS Encryption Algorithm

UIA – UMTS Integrity Algorithm

UICC – Universal Integrated Circuit Card

ULE – Unidirectional Lightweight Encapsulation

ULE Sec_ID – ULE Security Identifier

UMTS – Universal Mobile Telecommunications System

UTRAN – UMTS Terrstrial Radio Access Network

USAT – USIM Application Toolkit

USIM – Universal Subscriber Identity Module

UUI – Unsecured ULE Information

VCM – Variable Coding and Modulation

VLR – Visitor Location Register

VoIP – Voice over IP

VPN – Virtual Private Network

WiMAX – Worldwide Interoperability for Microwave Access

XRES – Expected Response

Résumé en français

Dans ce travail de thèse, nous traitons la problématique de la sécurité des données échangées basée sur les séquences chaotiques à savoir: la sécurité des communications IP par DVB satellitaire, la sécurité de l'UMTS et l'intégrité des images JPEG par tatouage fragile.
D'abord, le problème de la sécurité des communications unicast et multicast par IP via DVB-S est traité. A ce sujet, nous proposons une nouvelle solution de sécurité basée sur le chiffrement des paquets IP transportés et aussi, le chiffrement du code MAC associé. La solution proposée, s'appuie sur: une gestion des clés multicouche, des fonctions chaotiques pour la génération des clés et pour le chiffrement, un PDU spécifique pour le transport des clés, et un message d'alarme pour rétablir la synchronisation entre le fournisseur et le client.
Ensuite, la question de la sécurité de l'UMTS est analysée et améliorée. L'aspect le plus sensible est l'accès sécurisé au réseau. A ce sujet, nous proposons: une amélioration de l'identification des utilisateurs, en protégeant d'avantage l'identité permanente; une protection de la clé secrète K contre les attaques cryptographiques, en chiffrant les messages de la procédure AKA et en utilisant une valeur temporaire de la clé K; et des changements dans la procédure de choix des algorithmes de chiffrement et d'authenticité.
Enfin, un nouvel algorithme efficace de tatouage numérique fragile basé chaos, pour la vérification de l'intégrité des images JPEG est réalisé. La conception de l'algorithme, découle de la cryptanalyse effectuée sur l'algorithme de Wang 2008. La méthode de cryptanalyse est ensuite modélisée par des chaînes de Markov du premier ordre.

Titre et résumé en Anglais

Communication protocols secured with chaotic sequences.
Applications for: IP over DVB-S and the UMTS.

In this thesis we have studied new ways of using chaotic functions to ensure information security. Therefore, we have addressed three themes of research: the security of IP communications over satellite DVB, UMTS security and digital watermarking.
Firstly we study the security of unicast and multicast IP communications over satellite DVB. We propose a new security solution for this type of communications that encrypts the IP packet and MAC code and that protects the authenticity and integrity of the ULE header and of the IP packet. This solution uses a multi layer key management system, chaotic functions for the encryption of the data and the generation of the secret keys, a customized PDU for the transport of the keys and an alarm message to restore the synchronization between the ISP and the client.
We analyze and propose improvements for the security of the UMTS. The network access is at the heart of UMTS security. The enhancements we propose are: user identification using an improved protocol that ensures the protection of: the permanent identity, the secret key K against cryptographic attacks using a temporary key and the encryption of the messages. The modified protocols for security algorithms negotiation and TMSI updating that make the choices of the serving network visible to the users.
Finally, we address the information integrity of JPEG images and we propose a new chaos based fragile watermarking algorithm that is efficient and robust. This algorithm is the result of the cryptanalysis that we have developed against the watermarking algorithm proposed by Wang in 2008. In addition we have also simulated the cryptanalysis using first order Markov chains.

Mots clés : Sécurité de l'information, Séquence chaotique, IP via DVB-S, UMTS, Gestion des clés, Tatouage fragile basé chaos.

Discipline : Electronique